ON THE GENERALIZED OGG CONJECTURE

Abstract. We prove that, for every positive integer N, every rational torsion point of $J_0(N)$ is represented by a rational divisor of degree zero supported on the cusps.

Contents

1. Introduction 3
1.1. Previous results 3
1.2. The proof 4
2. Divisors and congruence covers 4
2.1. Correspondences and primary projections 6
2.2. Integral models 6
2.3. Integral cusp expansions 6
2.4. Cyclic divisor descent 8
2.5. Nilpotent Picard classes 9
3. Two-primary torsion at odd levels 10
3.1. Integral differentials and the quotient 10
3.2. Support descent 11
3.3. The trace ideal 12
3.4. The cusp lattice and eta periods 14
3.5. The constant defect 17
3.6. The geometric etale socle 19
3.7. Compact characters and multiplicative roots 20
3.8. The exceptional coset 21
4. The Gaussian divisor kernel 22
4.1. A ramified cover and its half-class 22
4.2. Affine branch divisors 23
4.3. Compression to affine patterns 24
5. The semistable two-primary case 25
5.1. The nilpotent factor 25
5.2. An integral ordinary Frobenius quotient 26
5.3. Logarithmic differentiation 27
5.4. The ordinary socle 29
5.5. The ordinary Eisenstein index 30
5.6. The logarithmic lattice 31
5.7. The ordinary torsion group 32
6. The level $4M$ 33
6.1. The symmetric-group diagram 33
6.2. A cubic model for the Prym 34
6.3. The first Gaussian correction 35
6.4. Rational eta units and norms 36
6.5. The second Gaussian correction 39
6.6. The three-old quotient 40
7. The dyadic induction 40

Date: November 26, 2026.

hexagon:2611.01109v1

7.1. The regular fine model 40
7.2. Ordinary charts and the Picard identity 42
7.3. The based quadratic cover 43
7.4. Norms at higher dyadic levels 44
7.5. Invariant divisors and the two descents 47
8. The prime three 48
8.1. Semistable levels 48
8.2. The cubic quotient at level $9M$ 50
8.3. The base case at $9M$ 53
8.4. Rational unit norms 57
8.5. Cyclotomic unit norms 61
8.6. The cyclic quotients and integral model 65
8.7. Formal subgroups and fixed branches 69
8.8. The balanced operator 71
8.9. The corrected branch equation 73
8.10. Descent and induction 75
9. The primary parts at primes at least five 77
9.1. The marked arithmetic tower 77
9.2. A saturated quadratic descent at the lower level 79
9.3. The integral full-p seed 80
9.4. The two surviving formal depths 81
9.5. Primitive depth and one global compact cover 83
9.6. Cuspidality of the marked seed 85
9.7. The whole higher-depth model 85
9.8. The ordinary fiber and the balanced cocycle 87
9.9. The compact character remainder 88
9.10. Arithmetic cusp orbits and shifted eta units 89
9.11. The vertical norm and all shifted congruences 91
9.12. Two descents and the full marked induction 93
10. Assembly at every level 94
Appendix A. Exact computational certificates 95
A.1. Odd-level residue matrices 97
A.2. Dyadic models and unit norms 105
A.3. Odd-prime group rings and marked units 188
A.4. Ternary geometry identities 258
A.5. Ternary Prym and auxiliary-prime identities 260
A.6. Ternary units and complete norm lattices 265
References 303

1. Introduction

For a positive integer N, the noncuspidal points of $X_0(N)$ classify elliptic curves equipped with a cyclic subgroup of order N, and $J_0(N) = \mathrm{Jac}(X_0(N))$ is its Jacobian. Noncuspidal rational points on $X_0(N)$ describe cyclic isogenies over $\mathbf{Q}$; those on $X_1(N)$ describe elliptic curves with a specified rational point of order N. These interpretations connect questions about elliptic curves to the arithmetic of modular curves. The cusps supply rational base points and explicit divisor classes in their Jacobians. Ogg's work on these classes formed part of the development that led to Mazur's theorems on rational torsion and Eisenstein ideals [36, 37, 34]; see also the historical account [1].

In 1973, Ogg computed the order of the class $[(0) - (\infty)]$ on $J_0(\ell)$ for a prime $\ell > 3$: it is the numerator of $(\ell - 1)/12$ [36, §3, pp. 227–230]. His calculation uses modular units obtained from the discriminant and the Dedekind eta function. The same paper relates cuspidal classes to unramified intermediate covers of $X_1(\ell) \to X_0(\ell)$ and records the groups generated by cusps at the genus-one levels. Ogg subsequently conjectured that $[(0) - (\infty)]$ generates all of $J_0(\ell)(\mathbf{Q})_{\mathrm{tors}}$ [37, Conjecture 2]. Mazur proved this conjecture in 1977: for every prime $\ell \geq 5$, the rational torsion group is cyclic of the stated order, with this generator [34, Theorem (1)]. His Eisenstein ideal compares the Hecke action on cuspidal divisors, modular forms, and torsion Galois representations. These comparisons remain central at composite level, where the operators at primes dividing the level and the fields of definition of the cusps also enter.

Let $C_{\mathrm{rat}}(N)$ be the image in $J_0(N)(\mathbf{Q})$ of the degree-zero divisors defined over $\mathbf{Q}$ and supported on the cusps. Let $C_{\mathrm{geom}}(N) \subset J_0(N)(\overline{\mathbf{Q}})$ be the image of all geometric degree-zero cuspidal divisors, and put

$$C_{\mathrm{geom}}(N)(\mathbf{Q}) = C_{\mathrm{geom}}(N) \cap J_0(N)(\mathbf{Q}).$$

The Manin–Drinfeld theorem [33, 15] gives

$$C_{\mathrm{rat}}(N) \subseteq C_{\mathrm{geom}}(N)(\mathbf{Q}) \subseteq J_0(N)(\mathbf{Q})_{\mathrm{tors}}.$$

The generalized Ogg conjecture identifies the rational torsion with the rational cuspidal classes. The equality of the two cuspidal groups asks, in addition, for rational cuspidal divisor representatives. Motivated by a question of Ribet, Yoo formulated this assertion in [56, Conjecture 1.3]. We prove both equalities at every level.

Theorem 1.1. *For every positive integer N,*

$$J_0(N)(\mathbf{Q})_{\mathrm{tors}} = C_{\mathrm{rat}}(N) = C_{\mathrm{geom}}(N)(\mathbf{Q}).$$

1.1. **Previous results.** The extension from prime to prime-power level brought the component groups of Néron models into the calculation. For $N = \ell^r$, with $r \geq 2$ and $\ell \geq 5$ a prime satisfying $\ell \not\equiv 11 \pmod{12}$, Lorenzini determined the prime-to-2ℓ part of the rational torsion and identified it with the corresponding cuspidal part [29]. The proof uses specialization to component groups and explicit relations among cusps. Ling subsequently computed rational cuspidal subgroups and related component groups at odd prime-power levels [27]. These calculations made the dependence on the exponent of a prime in the level explicit.

Takagi computed the cuspidal class number at squarefree level [49]; this formula is an input to Ohta's comparison. At squarefree level, Ohta proved the equality of rational torsion and rational cuspidal groups on the p-primary part for every $p \geq 5$, and for $p = 3$ when $3 \nmid N$ [38]. For general N, Ren proved that the rational torsion has trivial p-primary part when $p \nmid 6N \prod_{\ell \mid N}(\ell^2 - 1)$ [41]. Ribet and Wake gave a proof for squarefree N and $p \nmid 6N$ by comparing annihilators as modules over the Hecke algebra [45]. Their argument relates the torsion comparison directly to the Eisenstein ideal and its action on modular forms.

Yoo treated arbitrary levels at every prime $p \geq 5$ satisfying $p^2 \nmid N$ [55, Theorem 1.4]. The same theorem gives the 3-primary equality when $9 \nmid N$ and either $3 \nmid N$ or some prime divisor

of N is congruent to -1 modulo 3. Thus the known result at $p \geq 5$ includes all multiplicities of the other prime factors of N. We use precisely these statements, recorded in Theorem 2.1. The present argument treats the entire 2-primary part, the additional 3-primary cases, and the p-primary parts with $p \geq 5$ and $p^2 \mid N$.

Explicit calculations have also settled individual composite levels. Lupoian computed two-torsion through theta hyperplanes and verified the generalized Ogg conjecture for $N = 42, 55, 63, 72, 75$ [30, Theorem 1].

The rationality of a cuspidal divisor representative has a parallel history. Wang and Yang proved $C_{\mathrm{rat}}(N) = C_{\mathrm{geom}}(N)(\mathbf{Q})$ for $N = n^2M$, where $n \mid 24$ and M is squarefree [52]. Guo, Yang, Yoo, and Yu proved this equality for $N = \ell^2 M$, with ℓ any prime and M squarefree, including $\ell \mid M$ [22]. Their work develops modular-unit descriptions adapted to the Galois orbits of cusps. Yoo computed the structure of $C_{\mathrm{rat}}(N)$ for general N [56]. More recently, Yoo and Yu proved the equality of the two cuspidal groups for $N = ML^2$, with M squarefree and L an odd prime power or the product of two odd prime powers [57]. In the present proof, the construction of rational cuspidal divisors for every rational torsion point gives both equalities in Theorem 1.1.

1.2. **The proof.** We first relate multiplicative torsion to congruence covers. We construct a finite flat subgroup of the Néron model and a based cover with integral expansions at every finite place. The unbounded-denominators theorem of Calegari, Dimitrov, and Tang [4] makes this cover congruence. Wohlfahrt's theorem [53] then identifies its level from its compact cusp widths. The use of integral root expansions in this argument appears in Yin [54, Proposition 8.1]; Chi, Liu, and Nie apply the method to isogeny lifts and the Manin constant [7, §4]. Najman uses the common-cover construction and a characteristic-two Hecke-socle comparison for rational torsion on $J_1(p)$ [35, §§3, 5]. We give the integral subgroup and descent constructions needed here in Section 2 and at their later uses.

For the 2-primary part at odd level and at level $2M$ with M odd, an integral Eisenstein-index calculation and the Hecke socle bound compare torsion of every order with cuspidal divisors. The odd-level calculation also gives a relation theorem for the closed Gaussian points in the rational Picard group. This theorem supplies the two divisor corrections in the passage from $4M$ to $2M$.

At higher prime-power depth, the proof uses two degeneracy maps on a common integral model. We compute their action on the whole special fiber, including its multiplicities, and obtain balanced identities for torsion classes. The subsequent descent requires equality between function norms and cusp-unit norms, with constants included. We prove this equality by expanding eta products at the cusps and solving the resulting integer congruences. Hilbert 90 first controls the divisors fixed by a deck transformation and then descends a divisor class to the lower curve. The induction at 3 includes all $\mathbf{Q}(\zeta_3)$-rational primary torsion. At $p \geq 5$ it includes the scalar-fixed primary torsion on the marked curve. Field traces of the resulting cuspidal divisors give the required rational representatives.

The proof ends in Section 10 with the primary decomposition of a rational torsion point. The appendix contains the finite matrix, phase, group-ring, and norm calculations, with their complete inputs, programs, and outputs.

2. Divisors and congruence covers

For $N \geq 1$, the congruence subgroups used below are

$$\Gamma(N) = \ker\bigl(\mathrm{SL}_2(\mathbf{Z}) \longrightarrow \mathrm{SL}_2(\mathbf{Z}/N\mathbf{Z})\bigr),$$

$$\Gamma_1(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbf{Z}) : c \equiv 0,\ a \equiv d \equiv 1 \pmod{N} \right\},$$

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbf{Z}) : c \equiv 0 \pmod{N} \right\}.$$

The map in the first line is reduction modulo N.

For a smooth compact modular curve X over a number field F, write $J_X = \mathrm{Jac}(X)$ and let $\mathrm{Div}^0_{\mathrm{cusp}}(X_{\overline{F}})$ be the group of geometric degree-zero divisors supported on its cusps. Put

$$C_F(X) = \mathrm{im}\big(\mathrm{Div}^0_{\mathrm{cusp}}(X_{\overline{F}})^{\mathrm{Gal}(\overline{F}/F)} \longrightarrow J_X(F)\big).$$

For $X/\mathbf{Q}$, write $C_{\mathrm{rat}}(X) = C_{\mathbf{Q}}(X)$. Throughout, $X_0(N)$ is the smooth compact modular curve over $\mathbf{Q}$, and we abbreviate $J_N = J_{X_0(N)}$, $C_F(N) = C_F(X_0(N)_F)$, and $C_{\mathrm{rat}}(N) = C_{\mathbf{Q}}(N)$. The cusp ∞ defines the Abel–Jacobi map $x \mapsto [x - \infty]$. Write $C_{\mathrm{geom}}(N)$ for the image of all geometric degree-zero cuspidal divisors; $C_{\mathrm{geom}}(N)(F)$ denotes its $\mathrm{Gal}(\overline{F}/F)$-fixed subgroup. Put

$$\Sigma(N) = \ker\big(J_0(N) \longrightarrow J_1(N)\big).$$

Here $J_0(N) = J_N$, $J_1(N) = \mathrm{Jac}(X_1(N))$, and the map is pullback on Picard varieties. The character realization of $\Sigma(N)$ uses the maximal compact unramified quotient of $X_1(N) \to X_0(N)$. Manin–Drinfeld gives the torsion of every cuspidal divisor class [33, 15]. Consequently

$$C_{\mathrm{rat}}(N) \subseteq C_{\mathrm{geom}}(N)(\mathbf{Q}) \subseteq J_N(\mathbf{Q})_{\mathrm{tors}}. \tag{2.1}$$

For a positive integer n, we use the arithmetic functions

$$\tau(n) = \#\{d > 0 : d \mid n\}, \qquad \sigma_1(n) = \sum_{d \mid n} d, \qquad \varphi(n) = \#(\mathbf{Z}/n\mathbf{Z})^\times,$$

$$\psi(n) = n \prod_{p \mid n} (1 + p^{-1}), \qquad \omega(n) = \#\{p \text{ prime} : p \mid n\}.$$

The valuation v_p on $\mathbf{Q}$ is normalized by $v_p(p) = 1$. Write $n_p = p^{v_p(n)}$, $\mathrm{oddpart}(n) = n/2^{v_2(n)}$, and $\mathrm{num}(a/b) = a$ for positive coprime integers a, b. For a finite group G, $|G|_p$ is the p-part of its order; for an abelian group A, $A[p^\infty]$ is its p-primary torsion subgroup. For an abelian variety or a p-divisible group A over a field F, write

$$T_p A = \varprojlim_r A[p^r](\overline{F}),$$

with transition maps $[p]$. For a group over a discrete valuation ring of mixed characteristic, this notation refers to its generic fiber. For a number field F, $\mathcal{O}_F$ is its ring of integers, and $\mathcal{O}_{F,v}$ is the valuation ring of its completion F_v at a finite place. Put $\zeta_m = e^{2\pi i/m}$ and

$$\Phi_m(X) = \prod_{\substack{1 \le a \le m \\ \gcd(a,m)=1}} (X - \zeta_m^a).$$

Write μ_m for the group scheme defined by $T^m = 1$, with multiplication as its group law, and $\mu_{p^\infty} = \varinjlim_r \mu_{p^r}$. Define the cyclotomic characters by

$$\sigma\zeta_m = \zeta_m^{\chi_m(\sigma)}, \qquad \chi_m : \mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q}) \longrightarrow (\mathbf{Z}/m\mathbf{Z})^\times, \qquad \chi_{p^\infty} = \varprojlim_r \chi_{p^r}.$$

For z in the upper half-plane and $q = e^{2\pi i z}$, put

$$\eta(z) = q^{1/24} \prod_{n \ge 1} (1 - q^n), \qquad \Delta(z) = \eta(z)^{24}, \qquad E_2(z) = 1 - 24 \sum_{n \ge 1} \sigma_1(n) q^n.$$

Theorem 2.1 (Yoo, Theorem 1.4)**.** *Let ℓ be a prime. If $\ell \ge 5$ and $\ell^2 \nmid N$, then $J_N(\mathbf{Q})[\ell^\infty] = C_{\mathrm{rat}}(N)[\ell^\infty]$. For $\ell = 3$, the same conclusion holds if $9 \nmid N$ and either $3 \nmid N$ or some prime $q \mid N$ satisfies $q \equiv -1 \pmod 3$.*

This is [55, Theorem 1.4]. The remaining three-primary cases at depth at most one will be proved below by an auxiliary-prime descent.

2.1. **Correspondences and primary projections.** For a correspondence with maps α, β, its Picard action is $\beta_*\alpha^*$. Its action on logarithmic differentials is $\mathrm{Tr}_\beta\, \alpha^*$; under a principal polarization this is the Rosati-transposed cotangent action. We retain this convention at bad primes as well as good ones. For a finite map f of degree d, $f_*f^* = [d]$ on the lower Jacobian.

Whenever an integer prime to ℓ is inverted on a finite ℓ-primary group, the inverse means an integer inverse modulo the exponent of that group. If a descent gives a point R with mR torsion, then R is torsion and has primary projections.

2.2. **Integral models.** Katz–Mazur's cyclic moduli problem, together with a representable finite étale auxiliary level prime to the residue characteristic, has a regular two-dimensional model. Its compactification is the normalization of the same j-line in its generic function field [25, Theorems 6.6.1–6.6.2 and 8.6.2–8.6.3]. At a compact cusp, after a finite unramified coefficient extension, the normalization has a cyclotomic coefficient discrete valuation ring (DVR) and a Tate parameter. At the coarse cusp ∞ of $X_0(N)$, at every prime, this parameter is the unscaled q and the complete ring is $\mathbf{Z}_\ell[[q]]$ [25, 8.11.10][17, proof of Proposition 2]. The fine constructions below specify the auxiliary level, its connected component, and the transported marking.

Let R be an unramified complete DVR with perfect residue field, and let $\mathscr{X}/R$ be a proper flat regular model of a geometrically connected smooth curve. Suppose it has a smooth section meeting a component of multiplicity one. The gcd of the geometric component multiplicities is then one. Raynaud's cohomological-flatness criterion and Picard comparison give

$$\mathrm{Pic}^0_{\mathscr{X}/R} \simeq \mathscr{J}^0, \tag{2.2}$$

where $\mathscr{J}$ is the Néron model over R [39, Theorems 7.2.1, 8.1.4, and 8.2.1]. The identity functor is smooth and separated. One description of the separatedness assertion is useful. A generically trivial line bundle of degree zero on every component is given by a vertical divisor V with $V \cdot C = 0$ for every component C. The kernel of the intersection matrix is generated over $\mathbf{Q}$ by the whole fiber F. Thus $V = aF$; the multiplicity-one component forces $a \in \mathbf{Z}$, and aF is principal. After a ramified extension, we use the base change of (2.2) and of this Néron model.

For a finite flat commutative group H, Raynaud's Picard–torsor duality identifies homomorphisms $H \to \mathrm{Pic}_{\mathscr{X}/R}$ with the corresponding H^D-torsors, modulo torsors from the base [39, Proposition 6.2.1]; here H^D is the Cartier dual. Its hypotheses are properness, flatness, finite presentation, and universal $H^0(\mathcal{O}) = R$. Rigidifying at the chosen section removes the base torsor. If H^D is étale, the resulting based torsor is finite étale over the whole model. Its generic restriction is the based torsor associated with $H_L \to \mathrm{Pic}_{\mathscr{X}_L/L}$, where $L = \mathrm{Frac}(R)$.

2.3. **Integral cusp expansions.** Applications of the unbounded-denominators theorem to congruence covers appear in [54, 7, 35]. We use the following number-field form.

Lemma 2.2. *Let X be a compact congruence modular curve over a number field L, with its fixed modular function j, and let $Y \to X$ be a geometrically connected finite cover unramified on the compact curves. Fix a cusp of Y and its unscaled modular parameter t, after enlarging L to contain its coefficient field. Suppose that, for every finite place v of L, expansion on this same based branch gives an embedding*

$$L(Y) \hookrightarrow \mathrm{Frac}(\mathcal{O}_{L,v}[[t]]) \tag{2.3}$$

compatible with one Laurent expansion over L, and that $j(t) \in \mathcal{O}_{L,v}[[t]][t^{-1}]$. Then the analytic subgroup defining Y is a congruence subgroup.

Proof. Choose a primitive element θ for the finite separable extension $L(Y)/L(j)$. In its monic equation, multiply θ by one polynomial $c(j) \in \mathcal{O}_L[j] \setminus \{0\}$ clearing all denominators of all coefficients. The new element $z = c(j)\theta$ is still primitive and is integral over $\mathcal{O}_L[j]$. For example, a coefficient $a_i(j)$ becomes $c(j)^i a_i(j)$, which is integral because already $c(j)a_i(j)$ is.

Fix v. The image $z(t)$ is integral over the normal ring $\mathcal{O}_{L,v}[[t]][t^{-1}]$. Hypothesis (2.3) puts it in that ring's fraction field. Normality therefore gives

$$z(t) \in \mathcal{O}_{L,v}[[t]][t^{-1}].$$

The function and branch have been fixed before v was chosen, so their Laurent coefficients lie in L and are integral at every finite place. They lie in $\mathcal{O}_L$. There is one finite pole order, namely the pole order of the fixed rational function at the chosen cusp.

The integrality of z over $L[j]$ also shows that it has no poles away from the cusps. Indeed, at a noncuspidal valuation a negative value of z would make the leading term of its monic equation the unique term of smallest valuation. Choose an integer $a \geq 0$ such that $\Delta^a z$ is holomorphic at all the cusps. It is a holomorphic modular form of weight $12a$ for the subgroup defining Y, with integral coefficients in a fractional power of the modular q. The unbounded-denominators theorem of Calegari–Dimitrov–Tang, including its number-field formulation [4, Theorem 1.0.1 and Remarks 6.3.1–6.3.2], implies that this form is congruence modular. Hence z is fixed by a principal congruence subgroup. Since $L(Y) = L(j, z)$ and j is invariant under the full modular group, that subgroup fixes the entire function field. It is therefore contained in the subgroup defining Y. □

An integral étale neighborhood of the based cusp or an integral root on the fixed line-bundle cover will give (2.3) in the original cusp parameter.

Theorem 2.3. *Let $\varphi : A \to J_N$ be a $\mathbf{Q}$-isogeny whose morphism on the original Néron formal groups is an isomorphism at every finite prime. The based cover*

$$Y = X_0(N) \times_{J_N} A \longrightarrow X_0(N), \qquad y_\infty = (\infty, 0),$$

is geometrically connected and congruence. Under Jacobian autoduality, the kernel of the dual isogeny is contained in $\Sigma(N)$.

Proof. If $J_N = 0$, then $A = 0$ and $Y = X_0(N)$, which gives the assertions. Otherwise the Abel–Jacobi map identifies $H_1(X_0(N)(\mathbf{C}), \mathbf{Z})$ with the period lattice of its Jacobian. It is in particular surjective on the abelianized fundamental group. An isogeny corresponds analytically to a finite-index sublattice. Its pullback along this surjection has transitive monodromy, so $Y_{\mathbf{C}}$ is connected. The cover is finite étale on the compact curves because the isogeny is étale in characteristic zero.

At a prime ℓ, extend Abel–Jacobi from the smooth locus of the integral modular curve to the original Néron model. The cusp section maps to zero. The invertible tangent matrix of φ makes the Néron morphism étale at zero. In the integral fiber product, the local ring B at $(\overline{\infty}, 0)$ is a regular two-dimensional local domain and

$$\widehat{B} = \mathbf{Z}_\ell[[q]].$$

Here an étale local map with unchanged residue field induces an isomorphism on completed local rings. The injection $B \hookrightarrow \widehat{B}$ follows from Krull intersection. The generic fiber of this neighborhood is an open of the fixed geometrically integral $Y_{\mathbf{Q}_\ell}$. Thus its fraction field gives

$$\mathbf{Q}(Y) \hookrightarrow \mathbf{Q}_\ell(Y) \hookrightarrow \mathrm{Frac}(\mathbf{Z}_\ell[[q]]).$$

The branch is the one through y_∞ at every prime. The previous lemma proves that Y is congruence.

Let $\Gamma_Y \subseteq \Gamma_0(N)$ be its analytic subgroup, including $-I$. Compact étaleness preserves each cusp width. Conjugating the translation matrix at a cusp a/c gives the width $N/\gcd(c^2, N)$. All widths divide N and the width at zero is N. Wohlfahrt's level theorem therefore gives $\Gamma(N) \subseteq \Gamma_Y$ [53][31, Chapter II, Theorem 9]. The chosen cusp over ∞ has width one, so $T = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ is in Γ_Y. Modulo $\Gamma(N)$ this matrix generates the image of $\Gamma_1(N)$. Consequently $\pm\Gamma_1(N) \subseteq \Gamma_Y$, and the compact map from $X_1(N)$ factors through Y.

Write $i : X_0(N) \to J_N$, $j_Y : Y \to A$, and $\pi : Y \to X_0(N)$ for the maps in the pullback square. On Picard varieties $\pi^* i^* = j_Y^* \varphi^*$. Thus the dual kernel, identified with its image under i^*, is killed by pullback to Y, and hence by pullback to $X_1(N)$. This proves the Shimura containment. Geometric containment of the two $\mathbf{Q}$-subgroup schemes descends by faithful flatness. □

Theorem 2.4. *Let $G \subset J_N$ be a subgroup scheme over $\mathbf{Q}$ of order 2^r. If its schematic closure in the original Néron model over $\mathbf{Z}_2$ is a finite flat closed subgroup isomorphic to μ_{2^r}, then $G \subseteq \Sigma(N)$.*

Proof. Let $f : J_N \to B = J_N/G$ and let H be the stated closure. Write $\mathcal{J}_N$ and $\mathcal{B}$ for the Néron models over $\mathbf{Z}_2$. The extended map vanishes on H: it vanishes generically and H is flat, while the target is separated. For a group scheme G/R with identity section e, write $\omega_G = e^* \Omega^1_{G/R}$. The closed immersion $H \hookrightarrow \mathcal{J}_N$ gives a quotient of conormal modules

$$\operatorname{coker}(f^* \omega_{\mathcal{B}} \to \omega_{\mathcal{J}_N}) \twoheadrightarrow \omega_H = \mathbf{Z}_2/(2^r).$$

Hence the determinant length of $\operatorname{Lie} f$ is at least r. Raynaud's isogeny formula on the original Néron Lie lattices is

$$\operatorname{length} \operatorname{coker}(\operatorname{Lie} f) + \operatorname{length} \operatorname{coker}(\operatorname{Lie} f^t) = v_2(\deg f) = r \tag{2.4}$$

[40, Theorem 2.1.1] [5, §3.1 and (3.4.2)]. Both lengths are nonnegative, so $\operatorname{Lie} f^t$ is invertible over $\mathbf{Z}_2$. At an odd prime, a companion isogeny has composite $[2^r]$ with f^t. The product of the integral Lie matrices is a unit scalar, so each is invertible. Apply Theorem 2.3 to $f^t : B^t \to J_N$, using the principal polarization of J_N. Its dual kernel is exactly G. □

Corollary 2.5. *For every N,*

$$\ker\bigl(J_N(\mathbf{Q})_{\mathrm{tors}} \longrightarrow \mathcal{J}_N(\mathbf{F}_2)\bigr) \subseteq \Sigma(N)[2].$$

Reduction in this kernel is to the identity point.

Proof. In integral formal coordinates, a point reducing to zero has coordinates in $2\mathbf{Z}_2$. Multiplication by two sends them into $4\mathbf{Z}_2$. If a nonzero point has minimum coordinate valuation $s \geq 2$, its double has minimum valuation exactly $s+1$: the linear term has that valuation and all higher terms have valuation at least $2s > s+1$. Iterating cannot give zero. Odd multiplication is an invertible formal map. Formal rational torsion is therefore killed by two.

Let $P \neq 0$ be one such point. The two sections $0, P$ lie in an affine neighborhood of their common special point. Their scheme-theoretic union has coordinate ring an order in $\mathbf{Z}_2 \times \mathbf{Z}_2$ containing the diagonal. It is finite and free of rank two. Its immersion into the whole Néron model is closed, since it is proper over the base; the generic group law extends by flatness. Such an order has the form

$$R_a = \{(x, y) : x - y \in 2^a \mathbf{Z}_2\}, \qquad a \geq 1.$$

With $e = (0, 2^a)$, the split group law forces

$$\Delta(e) = e \otimes 1 + 1 \otimes e - 2^{1-a} e \otimes e.$$

Integrality forces $a \leq 1$. Thus $a = 1$, and $u = 1 - e$ satisfies $u^2 = 1$ and $\Delta(u) = u \otimes u$. The closure is μ_2. Theorem 2.4 now applies to the global subgroup generated by P. □

2.4. **Cyclic divisor descent.** Let $f : X \to Y$ be a cyclic map of degree m between smooth proper geometrically connected modular curves over a field K of characteristic zero, carrying cusps to cusps, with deck generator σ. Suppose $c \in X(K)$ is a totally ramified cusp, and put $b = f(c)$.

If a K-line bundle L has invariant class, choose an isomorphism $\sigma^* L \simeq L$ and scale it to act as the identity on L_c. The product of its m translates is an automorphism of L, hence a constant, whose value at c is one. This gives a linearization. At a ramified point, the tangent character is faithful on the inertia group. Twisting by an integral multiple of that point cancels the character

on the fiber of L. The chosen multiples are Galois compatible, so they form a K-divisor. After all characters are cancelled, the linearized bundle descends to Y. If the branch locus is cuspidal, the correction is a cusp divisor. If noncuspidal branch points remain, their correction must be retained; the Gaussian argument below treats exactly this situation for a double cover.

For the descent by functions, write $U_X(K)$ for the functions on X whose divisors are supported on cusps, including K^*. Assume ramification away from cusps is absent and

$$U_Y(K) \cap \operatorname{Nm} K(X)^* = \operatorname{Nm} U_X(K). \tag{2.5}$$

Let A be a degree-zero K-divisor, and suppose $(\sigma-1)[A]$ has a degree-zero cuspidal representative D. Choose f_0 with

$$(\sigma - 1)A - D = \operatorname{Div}(f_0).$$

Its norm is a cusp unit and a function norm. By (2.5), choose $g \in U_X(K)$ with $\operatorname{Nm} g = \operatorname{Nm} f_0$. Put $u = f_0/g$, so $\operatorname{Nm} u = 1$. Distinct automorphisms of a field are linearly independent over that field. Indeed, suppose $\sum_{i=1}^{r} c_i\tau_i(x) = 0$ for every x is a nonzero relation with the fewest terms, so all $c_i \neq 0$. A one-term relation is impossible on evaluating at $x = 1$, so $r \geq 2$. Choose y with $\tau_1(y) \neq \tau_r(y)$. Subtracting $\tau_r(y)$ times the relation at x from the relation at yx gives

$$\sum_{i=1}^{r-1} c_i(\tau_i(y) - \tau_r(y))\tau_i(x) = 0.$$

Its first coefficient is nonzero, contradicting minimality.

The cyclic Hilbert 90 formula is

$$t = \sum_{i=0}^{m-1} \left(\prod_{j=0}^{i-1} \sigma^j(u^{-1}) \right) \sigma^i(a), \qquad a \in K(X),$$

where an empty product is one. Linear independence of the distinct automorphisms $1, \sigma, \ldots, \sigma^{m-1}$ permits a choice of a with $t \neq 0$, since the coefficient of a is one. Reindexing gives

$$\sigma(t) = u \sum_{i=1}^{m} \left(\prod_{j=0}^{i-1} \sigma^j(u^{-1}) \right) \sigma^i(a) = ut.$$

Indeed, the term with $i = m$ equals the term with $i = 0$, because $\operatorname{Nm} u = 1$ and $\sigma^m = 1$. Thus $f_0/g = \sigma(t)/t$, and the noncuspidal part of $A - \operatorname{Div}(t)$ is invariant. At an unramified closed point, pullback has ramification index one, so an invariant integral divisor descends with integer coefficients, including at inert points.

We obtain $A - \operatorname{Div}(t) = f^*E + D'$ with D' cuspidal. Since $f^*b = mc$, replace E by $E - (\deg E)b$ and D' by $D' + m(\deg E)c$ to make both degrees zero. If $[A]$ is torsion, pushforward shows that $m[E]$ is torsion, because the correction is cuspidal. Hence $[E]$ is torsion. We may then take its primary projection.

2.5. **Nilpotent Picard classes.**

Lemma 2.6. *Let R be an unramified complete mixed-characteristic DVR with residue field k of characteristic p. Let $\mathscr{X}/R$ be proper, flat, and regular, with special fiber F satisfying $H^0(F, \mathcal{O}_F) = k$ and a smooth based section c on a multiplicity-one proper reduced component O with $H^0(O, \mathcal{O}_O) = k$. Let $r \geq 1$. Suppose a rigidified line bundle L on $\mathscr{X}$ has a compatible trivialization $L^{p^r} = \mathcal{O}$, and its special class is in the image of $H^1(F, 1 + I)$ for an ideal $I \subset \mathcal{O}_F$ with $I^p = 0$. Then at c, in compatible frames, the fixed power trivialization has the form*

$$t_c(q) = 1 + p^{r+1}a(q), \qquad a(q) \in R[[q]], \quad a(0) = 0.$$

Its p^r-th root with value one is integral in the same parameter q.

Proof. First $H^0(\mathscr{X} \bmod p^s, \mathcal{O}) = R/p^s$ for every s. The sequence $0 \to \mathcal{O}_F \xrightarrow{p^{s-1}} \mathcal{O}_{\mathscr{X} \bmod p^s} \to \mathcal{O}_{\mathscr{X} \bmod p^{s-1}} \to 0$ proves this by induction: subtract a lift of the scalar section at the preceding stage; the remainder is a scalar in k times p^{s-1}. The section c gives injectivity of the scalar map.

Choose frames whose reductions give transitions $1 + f_{ij}$ with $f_{ij} \in I$. Their transitions g_{ij} satisfy $g_{ij}^{p^r} \equiv 1 \pmod{p^r}$. Indeed $g_{ij}^{p} \equiv 1 \pmod{p}$ and raising an element of $1 + p^s R$ to the p-th power raises its congruence depth by at least one. The local power-trivialization units t_i therefore glue modulo p^r to a global scalar. Use the prescribed rigidification to make the frame at c have value one. This scalar is one, so $t_i = 1 + p^r \widetilde{h}_i$.

On the multiplicity-one component O, the regular functions

$$d_{ij} = \frac{g_{ij}^{p^r} - 1}{p^r}$$

vanish. To check this, at its generic DVR write $g_{ij} = 1 + pa$. Every binomial term after division has valuation at least $v_p\binom{p^r}{j} + j - r \geq j - v_p(j) \geq 1$. Generic vanishing extends on the reduced integral component O. Consequently the restrictions of $\widetilde{h}_i$ glue to a regular function on proper O. It is constant, and its value at c is zero. The completed special fiber at c is O, so $\widetilde{h}_c \in pR[[q]]$, as asserted.

The binomial root is well defined q-adically because $a(0) = 0$. For $j \geq 1$ its scalar coefficient has valuation

$$v_p\left(\binom{1/p^r}{j} p^{(r+1)j}\right) = j - v_p(j!) \geq 1.$$

Every factor $1 - ip^r$ in the numerator is a unit, which proves the equality. Only finitely many powers $a(q)^j$ contribute to each coefficient of q. The fixed root is therefore in $1 + pR[[q]]$. □

When this lemma is applied to a rational torsion point, the line-bundle cover is fixed first by quotienting the dual Jacobian by that point and dualizing. At primes different from p, the companion isogeny gives an integral formal inverse. The lemma supplies the same based branch at p. Lemma 2.2 and the compact-width argument then put the original point in $\Sigma(N)(\mathbf{Q})$. The cyclotomic character description of the Shimura group gives $\Sigma(N)(\mathbf{Q}) = \Sigma(N)[2]$ [28]; in particular its odd primary rational part is zero.

3. Two-primary torsion at odd levels

Throughout this section N is odd. Put $R = \mathbf{Z}_2$, $K = \mathbf{Q}_2$, and

$$J = J_0(N), \qquad C = C_{\mathrm{rat}}(N)[2^\infty], \qquad T_N = J(\mathbf{Q})[2^\infty], \qquad \Sigma = \ker\big(J_0(N) \longrightarrow J_1(N)\big).$$

The coefficients of a rational cusp divisor are constant on each rational cusp orbit. Fix an embedding $\overline{\mathbf{Q}} \hookrightarrow \overline{\mathbf{Q}}_2$; bars on torsion points denote reduction at 2 for this embedding.

Theorem 3.1. *For every odd positive integer N,*

$$J_0(N)(\mathbf{Q})[2^\infty] = C_{\mathrm{rat}}(N)[2^\infty].$$

The levels 1 and 3 have zero Jacobian and satisfy the theorem directly. We use the lattice and defect formulas below with $N > 1$. The proof also determines the special-fiber trace kernel used in the Gaussian argument.

3.1. **Integral differentials and the quotient.** We use the compactified coarse modular curve $X = X_0(N)_R$. For odd N it is smooth and proper with geometrically connected fibers, its reduced cusp divisor D_∞ is finite etale, and the completion at the cusp ∞ is $R[[t]]$, where $t = e^{2\pi i z}$ on the generic fiber; see [6, Proposition 6.4, Theorem 5.13, and the proof of Theorem 6.7].

Lemma 3.2. *A characteristic-zero weight-two form holomorphic at all cusps, whose complete expansion at ∞ is integral, defines a section of $\Omega^1_{X/R}(D_\infty)$. For cusp forms the analogous statement holds with $\Omega^1_{X/R}$. The resulting cusp-form lattice has injective reduction of its t-expansion modulo 2.*

Proof. The generic form gives a logarithmic differential on the coarse curve. Its possible horizontal poles are already accounted for by D_∞. Since the special fiber is geometrically integral, there is only one vertical prime divisor. A pole along this divisor is detected in the completed smooth cusp chart. The assumed integral expansion excludes such a pole. On the regular surface X, a rational section of an invertible sheaf with no codimension-one pole extends everywhere. This proves the first assertion and its cuspidal version.

For the last assertion, suppose an integral section has expansion zero modulo 2. Its reduction vanishes on the formal germ at a smooth point of the integral special curve, so it vanishes identically. The section is therefore divisible by 2 in the integral lattice. □

For $N \geq 5$ put $Y = X_1(N)_R$ and $\Delta_N = (\mathbf{Z}/N\mathbf{Z})^\times/\{\pm 1\}$. The finite modular map $\pi : Y \to X$ is finite flat of degree $|\Delta_N|$. Indeed, it is finite by the finite maps to the j-line. At a closed point the source is Cohen–Macaulay, while the target is regular of dimension two. A system of parameters on the target remains a system of parameters under a finite map, so the source is a maximal Cohen–Macaulay module over the target and is locally free. At codimension-one points the same conclusion follows from torsion-freeness over a discrete valuation ring.

Over $k = \overline{\mathbf{F}}_2$, a generic elliptic curve has automorphism group $\{\pm 1\}$. A generator of odd order distinguishes all the other diamonds. The diamond action on Y_k is consequently faithful and generically free. Hence

$$[k(Y_k) : k(Y_k)^{\Delta_N}] = |\Delta_N| = [k(Y_k) : k(X_k)].$$

The invariant function field is $k(X_k)$, and the finite comparison between the normal projective quotient and X_k is an isomorphism:

$$Y_k/\Delta_N = X_k. \tag{3.1}$$

Lemma 3.3. *Let a finite group G act on a smooth proper connected curve Y over an algebraically closed field k of characteristic p. Pullback from Y/G to Y is injective on the p-primary torsion points of its Picard group.*

Proof. If the pullback of a line bundle L is trivial, a trivialization converts its canonical G-linearization into a character $\rho : G \to k^\times$. If $L^{p^a} = 1$, the induced character satisfies $\rho^{p^a} = 1$. There is no nontrivial p-power root of unity in $k^\times$, so $\rho = 1$. The trivializing section is invariant and descends. More explicitly, a local trivialization of L gives

$$(\pi_*\pi^*L)^G = L \otimes (\pi_*\mathcal{O}_Y)^G = L.$$

The descended section trivializes L. □

Applied to (3.1), this proves that π^* is injective on all geometric special-fiber two-primary points. Thus the reduction of every point of $\Sigma[2]$ is zero.

3.2. **Support descent.**

Lemma 3.4. *Let q be an odd prime, let M be odd, and let $A = \mathbf{Z}/2^a\mathbf{Z}$. If a holomorphic Katz form at level qM has expansion $H(t^q)$, then there is a holomorphic form at level M with expansion $H(t)$. The assertion includes $q \mid M$ and every multiplicity of q in M.*

Proof. Starting with (E, C_{qM}), quotient by the standard order-q subgroup of C_{qM}. Write $\phi : E \to E'$ for this quotient and set $D_q = \ker(\phi^\vee)$. The resulting data are

$$(E', C_M, D_q).$$

When $q \mid M$, the two order-q lines D_q and $C_M[q]$ are transverse. Conversely, quotienting E' by D_q, taking the dual isogeny $\phi : E \to E'$, and taking $\phi^{-1}C_M$ recovers a cyclic group of order qM. To check cyclicity, work etale locally so that the relevant q-primary torsion is constant: choose the two transverse lines as coordinate axes in the free rank-two module. The preimage of the prescribed cyclic group is cyclic exactly when the added line is transverse. This description is unchanged when $v_q(M) > 1$.

The intermediate analytic group is

$$\Gamma' = \Gamma_0(M) \cap \Gamma^0(q) = \begin{pmatrix} 1 & 0 \\ 0 & q \end{pmatrix}^{-1} \Gamma_0(qM) \begin{pmatrix} 1 & 0 \\ 0 & q \end{pmatrix}.$$

Here $\Gamma^0(q)$ consists of the matrices in $\mathrm{SL}_2(\mathbf{Z})$ whose upper-right entry is divisible by q. Transport the form along this description, normalizing the invariant differential by the quotient isogeny. Its differential is a unit because q is invertible in A. On

$$E' = \mathrm{Tate}(t), \qquad C_M = \mu_M, \qquad D_q = \langle t^{1/q} \rangle,$$

the transported expansion is $H(t)$. The Tate description at the other cusps uses powers or roots of the cusp parameter and unit factors on differentials. Thus transport preserves holomorphy.

Put $L = \mathrm{lcm}(M, q)$ and pass to the full L-level cover, adjoining its roots of unity. If an additional prime-to-2 level is needed to make the moduli problem fine, add one and subsequently use faithfully flat descent. At a tested cusp write $s = t^{1/L}$. Translation

$$T_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

sends s to $\zeta_L s$ and fixes $H(s^L)$. This assertion is needed on each determinant component. On the component with Tate basis $(\zeta_L^u, t^{1/L})$, $u \in (\mathbf{Z}/L\mathbf{Z})^\times$, the translation replaces D_q by $\langle \zeta_q^u t^{1/q} \rangle$. Substitution in the original expansion gives

$$H\big((\zeta_q^u t^{1/q})^q\big) = H(t).$$

The q-expansion principle on each component therefore shows T_1-invariance. The proof of [9, Lemma 3.2] applies to each smooth connected full-level component, as discussed in [9, §1]. Over A, injectivity also follows from the field case by induction: a section with zero expansion is zero modulo 2, hence is twice a section modulo 2^{a-1}, to which induction applies.

It remains to identify the group generated by the descent data:

$$\langle \Gamma', T_1 \rangle = \Gamma_0(M). \tag{3.2}$$

The kernel of reduction modulo q is already in Γ'. If $q \mid M$, the image of Γ' is the diagonal torus and the image of $\Gamma_0(M)$ is the upper Borel; T_1 supplies the missing upper unipotent subgroup. If $q \nmid M$, these two images are the lower Borel and $\mathrm{SL}_2(\mathbf{F}_q)$, and the same upper unipotent subgroup generates the latter together with the former. Integral reduction and the Chinese remainder theorem give the stated images, including the conditions at all other prime powers in M. The determinant-changing diagonal transformations preserve the moduli data already used. Thus we have the complete descent datum for level M. Faithfully flat descent of sections on the elliptic locus, followed by the same tame Tate charts at the cusps, gives the required holomorphic form. □

3.3. **The trace ideal.** Let

$$E_K = \langle E_2(z) - dE_2(dz) : 1 < d \mid N \rangle_K, \qquad V = S_2(\Gamma_0(N), K) \oplus E_K,$$

where $S_2(\Gamma_0(N), K)$ is the space of weight-two cusp forms over K. For $f \in V$, write $f(t) = \sum_{n \geq 0} a_n(f) t^n$. Let $\widetilde{\mathbb{T}}$ be the full R-Hecke algebra on V, including every bad U_q, and let $\mathbb{T}$ be its cuspidal image. The ideal

$$I_{\mathrm{tr}} = (T_\ell - 1 - \ell : \ell \nmid N) \subset \widetilde{\mathbb{T}} \tag{3.3}$$

is generated over the primes $\ell \nmid N$ and includes $T_2 - 3$. We use the same notation for its image in $\mathbb{T}$.

For a Picard correspondence $u = \beta_* \alpha^*$, the action on differentials is $\mathrm{Tr}_\beta \, \alpha^*$. Under the Jacobian cotangent identification this is $(u^\dagger)^*$, where $u^\dagger$ denotes the Rosati adjoint. It is the convention for which the usual Fourier formulas, residues, and logarithmic derivatives commute. In particular $a_1(T_n f) = a_n(f)$, including the operators at bad primes.

Put

$$\mathcal{M}^+ = \{f \in V : a_n(f) \in R \text{ for all } n \geq 1\}.$$

The coefficient pairing gives

$$\mathcal{M}^+ = \mathrm{Hom}_R(\widetilde{\mathbb{T}}, R). \tag{3.4}$$

In characteristic zero the coefficient functionals span V^*: a form whose positive expansion vanishes is constant, and a constant weight-two form is zero. The relation $a_1 T_n = a_n$ and commutativity show that the pairing is nondegenerate on the Hecke side as well. Indeed, if $u \in \widetilde{\mathbb{T}}$ satisfies $a_1(uf) = 0$ for every f, then $a_n(uf) = a_1(uT_n f) = 0$ for every n, and $u = 0$. The operators T_n span the full algebra, so its R-dual imposes exactly the positive-coefficient integrality conditions. The algebra preserves an integral lattice and is a finite free, closed R-module. Reduction of (3.4) is therefore an injection into $t\mathbf{F}_2[[t]]$.

The same pairing gives

$$\mathbb{T} \simeq \mathrm{Hom}_R(\mathcal{S}, R), \qquad \mathcal{S} = H^0(X, \Omega^1_{X/R}). \tag{3.5}$$

For surjectivity, the coefficient functionals span the full dual modulo 2 by Lemma 3.2. Nakayama's lemma then gives the integral surjectivity. Injectivity is the same commutativity argument.

Lemma 3.5. *The series $F(t) = \sum_{n \geq 1} \sigma_1(n) t^n$ is not the positive expansion modulo 2 of an element of $\mathcal{M}^+$.*

Proof. Suppose otherwise, and choose $b \geq 0$ such that $H = 2^b f$ has integral constant coefficient c. Modulo 2^{b+1} its complete expansion is $c + 2^b F(t)$. For an odd $q \mid N$, the identity

$$U_q F(t) = (q+1)F(t) - qF(t^q)$$

gives

$$U_q H - (q+1)H = -q\{c + 2^b F(t^q)\} \pmod{2^{b+1}}.$$

Lemma 3.2, or its pullback to a fine odd level followed by Kodaira–Spencer, makes this an integral Katz form. Divide by the unit $-q$ and apply Lemma 3.4. This removes one occurrence of q from the level while retaining expansion $c + 2^b F(t)$. Repeating the operation exactly as often as the prime occurs in N reaches level one. A holomorphic weight-two form over $\mathbf{Z}/m\mathbf{Z}$ at level one has constant expansion [34, II, Proposition 5.6]. Its coefficient of t would nevertheless be $2^b \neq 0$ modulo 2^{b+1}, a contradiction. □

Lemma 3.6. *If $q^r \| N$, then*

$$U_q^{r-1}(U_q - 1)(U_q - q) \in I_{\mathrm{tr}}.$$

Proof. For each primitive cuspidal newform, use its two-adic Galois representation, and let its trace act scalarly on its full oldspace. On E_K use the trace of $1 \oplus \chi_{2^\infty}$. Chebotarev and closedness of $\widetilde{\mathbb{T}}$ show that the resulting operator $t(\sigma)$ is in this algebra and is congruent to $1 + \chi_{2^\infty}(\sigma)$ modulo I_{tr}. This follows by approximating the trace by good-prime Hecke operators in the finite collection of representations involved. Since q is odd, a Frobenius lift at q has $\chi_{2^\infty}(\sigma) = q$.

We verify the exact relation

$$U_q^{r+1} + qU_q^{r-1} = U_q^r t(\sigma) \tag{3.6}$$

on every oldspace. The standard newform and local representation facts used in this calculation are given in [44, Theorem 9.1.10, Sections 9.2 and 20.2]. If the conductor exponent at q is zero, the old basis $f_j = f(q^j z)$, $0 \le j \le r$, satisfies

$$U_q f_0 = a_q f_0 - q f_1, \qquad U_q f_j = f_{j-1} \quad (j \ge 1).$$

Its annihilating polynomial is $X^{r-1}(X^2 - a_q X + q)$, and the trace is a_q. If the conductor exponent is one, there are r old vectors, $U_q f_0 = \varepsilon f_0$ with $\varepsilon = \pm 1$, and the annihilator is $X^{r-1}(X - \varepsilon)$. The trace is $\varepsilon(1+q)$, and

$$X^2 - \varepsilon(1+q)X + q = (X - \varepsilon)(X - q\varepsilon).$$

If the conductor exponent is $e \ge 2$, trivial nebentypus gives $a_q = 0$. The old string is nilpotent of length at most $r - e + 1 \le r - 1$, so every term of (3.6) is zero. Finally, the formal $E_2(dz)$ strings have the first displayed matrix with $a_q = q + 1$; their modular hyperplane is stable and gives the same relation on E_K. Reduction of (3.6) modulo I_{tr} proves the assertion. □

Proposition 3.7. *The full trace ideal satisfies*

$$I_{\mathrm{tr}} = \mathrm{Ann}_{\widetilde{\mathbb{T}}}(E_K).$$

Proof. The inclusion from left to right follows from the good-prime Eisenstein eigenvalues. Modulo 2, the good-prime recurrences and Lemma 3.6 determine all positive coefficients from those indexed by divisors of N. At 2 the recurrence is $a_{2n} = a_n$; at an odd good prime it is the usual $a_{\ell n} = a_{n/\ell}$ modulo 2; and the bad-prime recurrence has order $r + 1$ at $q^r \| N$. The space of solutions is precisely

$$W = \langle F(t^d) : d \mid N \rangle_{\mathbf{F}_2}, \qquad \dim_{\mathbf{F}_2} W = \tau(N).$$

These series satisfy the recurrences, and their coefficients on the divisor indices form a triangular matrix with diagonal 1.

Set $A = \widetilde{\mathbb{T}}/I_{\mathrm{tr}}$. It surjects onto the image algebra on E_K, a free R-module of rank $\tau(N) - 1$. Its rational rank is the same: $T_2 - 3$ is zero on E_K and invertible on the cuspidal space, since every cuspidal eigenvalue satisfies $|a_2| \le 2\sqrt{2} < 3$ in characteristic zero. By (3.4), the dual of $A/2A$ embeds in W. If its dimension were $\tau(N)$, it would contain $F(t)$, contrary to Lemma 3.5. Thus A needs at most $\tau(N) - 1$ generators over R. A finite R-module of this rank with this generator bound has no torsion: in its decomposition into a free part and cyclic torsion modules, each nonzero torsion summand would add a generator modulo 2. The surjection to the Eisenstein image is consequently an isomorphism. Its kernel is the claimed annihilator. □

3.4. **The cusp lattice and eta periods.** For $d \mid N$ write

$$g_d = \gcd(d, N/d), \qquad w_d = \varphi(g_d).$$

The cusps of denominator d form one rational orbit of size w_d [56, Theorem 2.17 and Lemma 2.19]. The coefficient at each cusp in that orbit gives the lattice

$$\mathcal{D}_R = \{(a_d) \in R^{\{d|N\}} : \sum_{d|N} w_d a_d = 0\}. \tag{3.7}$$

It is primitive because $w_1 = 1$. For a vector $r = (r_\delta)$ with $\sum r_\delta = 0$, put

$$f_r = \frac{1}{24} \sum_{\delta|N} \delta r_\delta E_2(\delta z).$$

The residue at a cusp of denominator d is $(\Lambda_N r)_d$, where

$$24\Lambda_N(d,\delta) = A_N(d,\delta) = \frac{N \gcd(d,\delta)^2}{g_d d \delta}. \tag{3.8}$$

This follows either by the Tate parameter at that cusp or by the eta transformation formula: f_r is the logarithmic derivative of $\prod \eta(\delta z)^{r_\delta}$ wherever the latter is defined.

Both the weights and the matrix factor over prime powers. If $N = p^e$ and $d = p^a$, $\delta = p^b$, then

$$A_{p^e}(p^a, p^b) = p^{e+2\min(a,b)-a-b-\min(a,e-a)}.$$

Remove row factors $p^{e-a-\min(a,e-a)}$ and column factors p^{-b}. The remaining matrix is $(p^{2\min(a,b)})$. Replace row a by row a minus row $a-1$, in descending order of a. Its first diagonal entry is 1 and its later diagonal entries are $p^{2a} - p^{2a-2}$. Multiplying the removed factors back gives

$$\det A_{p^e} = p^{e(e-1)-\lfloor e^2/4 \rfloor}(p^2-1)^e. \tag{3.9}$$

Indeed, the exponent of p contributed by the removed factors and the differences is

$$\begin{aligned}\sum_{a=0}^{e}(e-a-\min(a,e-a)) - \sum_{b=0}^{e} b + \sum_{a=1}^{e}(2a-2) \\ = \frac{e(e+1)}{2} - \left\lfloor \frac{e^2}{4} \right\rfloor - \frac{e(e+1)}{2} + e(e-1) = e(e-1) - \left\lfloor \frac{e^2}{4} \right\rfloor.\end{aligned}$$

Here $\sum_{a=0}^{e} \min(a, e-a) = \lfloor e^2/4 \rfloor$. For general N, the tensor-product determinant gives

$$\det A_N = \prod_{p^e \| N} (\det A_{p^e})^{\tau(N)/(e+1)}.$$

Every weighted column sum is the same:

$$\sum_{d|N} w_d A_N(d, \delta) = \psi(N). \tag{3.10}$$

For an interior local column $1 \le b \le e-1$, the sum is

$$p^{e-b} + p^b + (1-p^{-1})\left(\sum_{a=1}^{b} p^{e+a-b} + \sum_{a=b+1}^{e-1} p^{e-a+b}\right) = p^e + p^{e-1}.$$

The two sums after multiplication by $1-p^{-1}$ are $p^e - p^{e-b}$ and $p^{e-1} - p^b$, respectively. For either endpoint column $b = 0, e$, the weighted sum is

$$p^e + 1 + (1-p^{-1})\sum_{a=1}^{e-1} p^a = p^e + 1 + (p^{e-1} - 1) = p^e + p^{e-1}.$$

Tensoring proves (3.10). Thus Λ_N identifies the sum-zero hyperplane with the weighted degree-zero hyperplane over K.

Let $\mathcal{M}$ impose integrality of the complete expansion in V, and set $\mathcal{E} = \mathcal{M} \cap E_K$ and $\mathcal{E}^+ = \mathcal{M}^+ \cap E_K$. The residue sequence on the smooth proper curve is surjective onto the trace-zero sections on its finite etale cusp divisor. Take its direct preimage over the submodule (3.7). Over K, subtracting the uniquely prescribed Eisenstein residue vector leaves a cusp form. Lemma 3.2 then identifies this preimage with $\mathcal{M}$, giving an exact sequence

$$0 \longrightarrow \mathcal{S} \longrightarrow \mathcal{M} \xrightarrow{\text{Res}} \mathcal{D}_R \longrightarrow 0. \tag{3.11}$$

The positive coefficients of f_r are

$$a_n(f_r) = -\sum_{\delta | \gcd(n,N)} \delta r_\delta \sigma_1(n/\delta).$$

On the indices $n \mid N$ the matrix is triangular with diagonal $-n$, a unit in R. Its constant coefficient is $24^{-1}\sum \delta r_\delta$. Hence, identifying Eisenstein forms with their r-vectors,

$$\mathcal{E}^+ = \{r \in R^{\{\delta | N\}} : \sum r_\delta = 0\}, \tag{3.12}$$

$$\mathcal{E} = \{r \in \mathcal{E}^+ : A(r) := \sum \delta r_\delta \in 8R\}. \tag{3.13}$$

Put $a = \min(3, \min_{p|N} v_2(p-1))$. Since the ideal generated by all $\delta - 1$, $\delta \mid N$, is generated by the $p-1$,

$$[\mathcal{E}^+ : \mathcal{E}] = 2^{3-a}. \tag{3.14}$$

Proposition 3.8. *The completed logarithmic-derivative lattice of rational modular units, modulo constants, is*

$$\mathcal{E}_{\mathrm{per}} = \{r \in \mathcal{E} : P_p(r) := \sum_{\delta|N} v_p(\delta) r_\delta \in 2R \text{ for every } p \mid N\}.$$

Moreover, the natural residue map identifies

$$\mathcal{M}/(\mathcal{S} + \mathcal{E}_{\mathrm{per}}) = C, \qquad \mathcal{E}/\mathcal{E}_{\mathrm{per}} = \Sigma[2] \subset C$$

as embedded Hecke modules.

Proof. First let u be a rational modular unit. Invert (3.8) over $\mathbf{Q}$ to obtain the rational vector r with $\Lambda_N r = \mathrm{Div}(u)$. After multiplying r by a sufficiently divisible positive integer m, the resulting eta product is a weight-zero modular function, with the same divisor as u^m. Their ratio is constant, and $d \log u = f_r$. On the good regular model, multiply u by a power of 2 to remove its vertical divisor. Its expansion near ∞ is $t^b U(t)$ with $U \in R[[t]]^\times$, so its logarithmic derivative is integral. The triangular coefficient formula forces every r_δ to belong to $\mathbf{Z}_2$. These exponents are rational, so their denominators are odd. Clear them by an odd integer D. The eta product with exponents Dr has logarithmic derivative equal to that of u^D, hence differs from u^D by a constant. It is consequently modular with trivial character.

For integer weight-zero eta products, Newman's criterion states that modularity on $\Gamma_1(N)$ is equivalent to

$$\sum \delta r_\delta \equiv 0 \pmod{24}, \qquad \sum (N/\delta) r_\delta \equiv 0 \pmod{24};$$

the character on $\Gamma_0(N)$ is the quadratic character of $\prod \delta^{r_\delta}$ [47, Theorem 1.1]. Applying the necessity and the character formula to Dr gives all the claimed parity conditions; D is odd, so it may be cancelled in these conditions.

Conversely, approximate an r satisfying the displayed conditions by integer sum-zero vectors to precision 2^j, $j \geq 3$. These retain $A = 0$ modulo 8 and every $P_p = 0$ modulo 2. As every divisor of N is odd,

$$B(r) := \sum (N/\delta) r_\delta \equiv N A(r) \pmod 8.$$

Multiply an approximation by an odd integer which is 1 modulo 2^j and 0 modulo 3. Both congruence sums are now divisible by 24, and the squareclass is a square. Newman's criterion gives a rational modular unit. The completed lattice generated by these units is closed, so approximation proves the asserted description of $\mathcal{E}_{\mathrm{per}}$.

The eta matrix has full rank; thus these units give a full-rank lattice of principal cusp divisors. If a rational cusp divisor is geometrically principal, choose a defining function f over a finite Galois extension $L/\mathbf{Q}$. For $\sigma \in \mathrm{Gal}(L/\mathbf{Q})$, the constants $c_\sigma = \sigma f / f$ satisfy $c_{\tau\sigma} = c_\tau \tau(c_\sigma)$. Choose $a \in L$ so that

$$b = \sum_{\sigma \in \mathrm{Gal}(L/\mathbf{Q})} c_\sigma^{-1} \sigma(a) \neq 0.$$

Such an a exists by independence of the automorphisms. Reindexing gives

$$\tau b = \sum_\sigma \tau(c_\sigma^{-1})(\tau\sigma)(a) = c_\tau \sum_\rho c_\rho^{-1} \rho(a) = c_\tau b.$$

Thus f/b is a rational defining function. The quotient of the rational degree-zero cusp lattice by these divisors is therefore the rational cuspidal group. Completion and (3.11) identify its two-primary part with $\mathcal{M}/(\mathcal{S} + \mathcal{E}_{\mathrm{per}})$.

It remains to identify the embedded subgroup. Let $x_p = P_p(r) \bmod 2$. The squareclass is $\prod p^{x_p}$. The compact-character description of the Shimura subgroup [28, Theorems 1 and 2] requires triviality at -1, at roots of d^2+1, at roots of d^2+d+1, and at the cusp inertia elements

with $(d-1)^2 = 0$ modulo N. For a quadratic character at odd level, the order-three condition and the cusp condition are automatic. The remaining conditions are

prime factors of N	condition
some $p \equiv 3 \pmod 4$	$\sum_{p\equiv 3 \ (4)} x_p = 0,$
all $p \equiv 1 \pmod 4$	$\sum_{p\equiv 5 \ (8)} x_p = 0.$

(3.15)

To see necessity for the parity image of $\mathcal{E}$, reduce $A(r) = \sum(\delta-1)r_\delta$ modulo 4 in the first case and modulo 8 in the second. For sufficiency set $r_p = x_p$, $r_1 = -\sum x_p$, and the other coordinates equal to zero. In the first case A is already divisible by 4; if it is 4 modulo 8, adding 2 to an r_p with $p \equiv 3 \pmod 4$ and subtracting 2 from r_1 corrects it. In the second case the stated condition already makes A divisible by 8.

Choose an integral representative and an odd Chinese-remainder multiple that enforces the two modulo-24 conditions. Its eta product F is a function on $X_1(N)$ with the prescribed quadratic character. The character conditions make it trivial on elliptic and cusp inertia. Therefore the divisor

$$D = \tfrac{1}{2}\operatorname{Div}(F^2)$$

on $X_0(N)$ is an integral rational cusp divisor: the orders are even before division, and the square-function on the base has rational expansion. Its pullback is $\operatorname{Div}(F)$, so $[D] \in \Sigma[2]$. The corresponding residue vector is the chosen r and thus gives the same class in C. If this class vanished, dividing F by a pullback from the base would leave a constant and its character would be trivial. Conversely, every admissible quadratic character was constructed in this way. This proves the embedded equality. Residues and logarithmic derivatives commute with the Picard correspondences, so the identification respects the Hecke action. □

In particular,

$$s := \dim_{\mathbf{F}_2} \Sigma[2] = \begin{cases} \omega(N), & p \equiv 1 \pmod 8 \text{ for every } p \mid N, \\ \omega(N) - 1, & \text{otherwise.} \end{cases} \tag{3.16}$$

The cyclotomic Galois action on Σ makes all points of $\Sigma[2]$ rational, consistently with their explicit divisor representatives.

3.5. **The constant defect.** Set

$$C^+ = \mathcal{M}^+/(\mathcal{S} + \mathcal{E}^+).$$

The lattice $\mathcal{E}^+$ is saturated in $\mathcal{M}^+$. Consequently $\mathcal{M}^+/\mathcal{E}^+$ is free, and the inclusion of $\mathcal{S}$ in it has finite cokernel C^+. Dualizing that inclusion over R and using (3.4)–(3.5) gives

$$(C^+)^\vee = \mathbb{T}/I_{\mathrm{tr}}, \qquad (\ )^\vee = \operatorname{Hom}(\ , \mathbf{Q}_2/\mathbf{Z}_2). \tag{3.17}$$

In detail, the dual of $\mathcal{M}^+/\mathcal{E}^+$ is the Eisenstein annihilator in $\widetilde{\mathbb{T}}$, the dual of $\mathcal{S}$ is $\mathbb{T}$, and the Ext^1_R term is $(C^+)^\vee$. Proposition 3.7 identifies its image ideal with I_{tr}.

Since $\mathcal{M} \cap (\mathcal{S} + \mathcal{E}^+) = \mathcal{S} + \mathcal{E}$, there is an exact sequence

$$0 \longrightarrow C/\Sigma[2] \longrightarrow C^+ \longrightarrow Q_N \longrightarrow 0, \qquad Q_N = \mathcal{M}^+/(\mathcal{M} + \mathcal{E}^+). \tag{3.18}$$

The constant coefficient embeds $\mathcal{M}^+/\mathcal{M}$ into K/R. Write its order as 2^b; then

$$|Q_N| = 2^{b-(3-a)}. \tag{3.19}$$

Lemma 3.9. *For odd $N > 1$, the group Q_N is cyclic of order 2 if every prime factor of N is 1 modulo 8, and is zero otherwise.*

Proof. First $b \leq 2$. If a positive-integral form has constant coefficient of denominator exactly 2^b, multiply by an integral unit and by 2^b to obtain an integral form with complete expansion 1 modulo 2^b. Lemma 3.4 removes each prime factor, with multiplicity, from the level. At level one, [34, II, Proposition 5.6] gives a constant expansion killed by 12. Thus $2^b \mid 12$.

If every prime factor is 1 modulo 4, then in fact $b \leq 1$. Suppose an integral characteristic-zero form H has expansion 1 modulo 4. Its reduction as a Katz form equals b_2, the weight-two Hasse square modulo 4. Indeed $u^2 b_2' = b_2 + 12r$ is the Weierstrass transformation rule, so b_2 defines such a form modulo 4, with Tate expansion 1; the componentwise expansion principle gives the equality [34, II, Sections 5.5–5.6].

Choose a root t of $t^6 - 36t^3 + 216 = 0$ in a finite extension of $\mathbf{Q}_2$ and consider

$$y^2 + txy + y = x^3. \tag{3.20}$$

The points of the Newton polygon are $(0,3)$, $(3,2)$, and $(6,0)$. Its lower hull is the segment of slope $-1/2$ joining the first and last points, so $v_2(t) = 1/2$. The Weierstrass coefficients give

$$b_2 = t^2, \qquad b_4 = t, \qquad b_6 = 1, \qquad b_8 = 0,$$
$$c_4 = b_2^2 - 24b_4 = t^4 - 24t,$$
$$c_6 = -b_2^3 + 36b_2b_4 - 216b_6 = -t^6 + 36t^3 - 216 = 0,$$
$$\Delta = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 = t^3 - 27.$$

Thus this curve has good reduction and $j = 1728$, while $v_2(b_2) = 1$. Each $p \mid N$ splits in $\mathbf{Z}[i]$; the action of i on $E[p^e]$ has two free rank-one eigenspaces over $\mathbf{Z}/p^e\mathbf{Z}$. Choosing one at each prime power gives a cyclic i-stable subgroup of order N, after a finite extension. The good model extends this prime-to-2 level structure. At this point weight two and the automorphism i force the value of H in characteristic zero to equal its negative, so that value is zero. Its reduction modulo 4 must equal b_2, whose valuation is exactly one. This is impossible.

If some $p \equiv 3 \pmod 4$, equation (3.14) gives $b \geq 2$, so $b = 2$ and $Q_N = 0$. If all primes are 1 modulo 4 and some is 5 modulo 8, it gives $b \geq 1$, so $b = 1$ and again $Q_N = 0$.

Finally suppose every prime is 1 modulo 8. Choose $p \mid N$ and put $n = \mathrm{num}((p-1)/12)$. The prime-level cusp order and Eisenstein index theorem [34, II, Proposition 9.7] give

$$|C_{\mathrm{rat}}(p)|_2 = n_2, \qquad |\Sigma(p)[2]| = 2, \qquad |\mathbb{T}(p)/I_{\mathrm{tr}}(p)|_2 = n_2.$$

Here the trace ideal is the Eisenstein annihilator by Proposition 3.7. Applying the already proved lattice identities at this prime level yields $|Q_p| = 2$. Since $\mathcal{E}^+ = \mathcal{E}$ there, a positive-integral form with half-integral constant exists. Pulling it up to level N keeps its Fourier coefficients and its membership in the sum of the cuspidal and principal-character Eisenstein spaces. Therefore $b \geq 1$. The upper bound gives $b = 1$; now $a = 3$ and (3.19) gives $|Q_N| = 2$. □

Combining the preceding results gives the full finite-index statement

$$0 \longrightarrow Q_N^\vee \longrightarrow \mathbb{T}/I_{\mathrm{tr}} \longrightarrow (C/\Sigma[2])^\vee \longrightarrow 0. \tag{3.21}$$

Adjoining primitive degree coordinates to the source and target lattices in (3.10) shows that the determinant of A_N on their degree-zero sublattices is $\det(A_N)/\psi(N)$. Indeed, in such bases the quotient coordinate is multiplication by $\psi(N)$. Equations (3.9), (3.14), and (3.16) therefore give

$$v_2|C| = \sum_{p^e \| N} \frac{\tau(N)e}{e+1} v_2(p^2-1) - \sum_{p \mid N} v_2(p+1) - 3(\tau(N)-1) + (3 - a + s). \tag{3.22}$$

The factors of p in (3.9) are odd; this explains why only $p^2 - 1$ contributes to the first sum.

3.6. **The geometric etale socle.** The logarithmic-differential argument below also occurs in [35, Lemma 5.2].

Lemma 3.10. *For every maximal ideal* $\mathfrak{m}$ *of* $\mathbb{T}$*,*

$$\dim_{\mathbb{T}/\mathfrak{m}} J_k(k)[\mathfrak{m}] \le 1, \qquad k = \overline{\mathbf{F}}_2.$$

Consequently the Pontryagin dual of each finite Hecke-stable subgroup of $J_k(k)[2^\infty]$ *is locally cyclic over* $\mathbb{T}$*.*

Proof. Smooth proper duality and base change give $\mathcal{S}/2\mathcal{S} = H^0(X_{\mathbf{F}_2}, \Omega^1)$. Equation (3.5) identifies this module with

$$\mathrm{Hom}_{\mathbf{F}_2}(\mathbb{T}/2\mathbb{T}, \mathbf{F}_2).$$

For any abelian variety B/k, the Weil pairing sends a point $P \in B[2](k)$ to a character $B^\vee[2] \to \mu_2$. Pulling back dz/z gives an injection

$$B[2](k) \otimes_{\mathbf{F}_2} k \hookrightarrow \omega_{B^\vee[2]} = \omega_{B^\vee}. \tag{3.23}$$

To check injectivity, over the perfect field k the reduced subgroup of $B[2]$ is an etale direct factor $(\mathbf{Z}/2)^f$, where $f = \dim_{\mathbf{F}_2} B[2](k)$. Its Cartier dual gives a direct factor μ_2^f in $B^\vee[2]$, whose coordinate logarithmic differentials are independent. The cotangent equality in (3.23) follows because [2] has zero differential.

Write i_B for this injection. For an endomorphism u of B, functoriality reads $i_B(uP) = (u^\vee)^* i_B(P)$. The principal polarization turns this into $(u^\dagger)^*$, precisely the Picard-correspondence action fixed above. Thus the injection is equivariant for every Hecke operator, including bad U_q. Equivalently one can conjugate the alternate cotangent convention by Fricke: its matrix conjugation exchanges $\mathrm{diag}(1, q)$ and $\mathrm{diag}(q, 1)$ and inverts diamonds. This equality of correspondences extends to the good abelian schemes.

For $\kappa = \mathbb{T}/\mathfrak{m}$, the $\mathfrak{m}$-socle of $\mathrm{Hom}_{\mathbf{F}_2}(\mathbb{T}/2\mathbb{T}, \mathbf{F}_2)$ is $\mathrm{Hom}_{\mathbf{F}_2}(\kappa, \mathbf{F}_2)$, of dimension one over κ. Tensoring with k and using (3.23) proves the bound. For a finite Hecke module A in the stated group, the dimension of $A^\vee/\mathfrak{m}A^\vee$ equals that of $A[\mathfrak{m}]$. Nakayama proves local cyclicity. □

Define the geometric trace kernel

$$\mathcal{K}_N = J_k(k)[2^\infty][I_{\mathrm{tr}}], \qquad \overline{C} = \mathrm{red}_2(C).$$

Corollary 2.5, Proposition 3.8, and Lemma 3.3 identify the kernel of reduction on C:

$$\overline{C} = C/\Sigma[2]. \tag{3.24}$$

The good Hecke operators act by $1 + \ell$ on rational cusp orbit divisors. This also follows by applying residues to the corresponding Eisenstein eigenforms and using the invertibility of Λ_N. Hence $\overline{C} \subset \mathcal{K}_N$. By Lemma 3.10 and (3.21),

$$|\mathcal{K}_N| \le |\mathbb{T}/I_{\mathrm{tr}}| = 2^{\epsilon_N} |\overline{C}|, \qquad \epsilon_N = \begin{cases} 1, & p \equiv 1 \pmod 8 \text{ for all } p \mid N, \\ 0, & \text{otherwise.} \end{cases} \tag{3.25}$$

The group is finite because its Hecke annihilator contains a power of 2. All its points are rational over $\mathbf{F}_2$. Frobenius F is invertible on geometric points, and Eichler–Shimura gives

$$0 = F(T_2 - 3) = (F - 1)(F - 2).$$

On a finite two-primary group $F - 2 = F(1 - 2F^{-1})$ is invertible: the inverse of the second factor is a finite geometric series. It follows that

$$\mathcal{K}_N = J(\mathbf{F}_2)[2^\infty][I_{\mathrm{tr}}]. \tag{3.26}$$

3.7. **Compact characters and multiplicative roots.** Suppose that every prime factor of an odd integer $M > 1$ is 1 modulo 4. Write $M = \prod q^{e_q}$ and $a_q = v_2(q-1) \geq 2$. Choose a generator θ_q of the local two-primary character group, of order 2^{a_q}. A two-primary character at q^{e_q} factors through q, since the reduction kernel has odd order.

Lemma 3.11. *Under the compact-character realization of $\Sigma(M)$, its two-primary geometric points correspond exactly to*

$$\left\{ (k_q) \in \prod_{q|M} \mathbf{Z}/2^{a_q}\mathbf{Z} : k_q \equiv 0 \pmod 2 \text{ for every } q, \quad \sum_q k_q/2 \equiv 0 \pmod 2 \right\}.$$

Proof. Choose a root i_q of -1 modulo q^{e_q}. For some odd b_q,

$$\theta_q(i_q) = i^{b_q}, \qquad \theta_q(-1) = -1.$$

At the Chinese-remainder root with orientation $\varepsilon = (\varepsilon_q)$, the character $\prod \theta_q^{k_q}$ has value

$$i^{\sum b_q k_q} (-1)^{\sum \varepsilon_q k_q}. \tag{3.27}$$

Triviality for orientations differing at only q forces k_q even. Once all k_q are even, the first factor of (3.27) is $(-1)^{\sum k_q/2}$, since b_q is odd. Thus triviality at all Gaussian inertia elements is equivalent to the two displayed conditions. These conditions also kill -1. The order-three inertia is killed by every two-primary character. The cusp inertia has diagonal entry 1 modulo each q, so its image in a two-primary local character is trivial. These are all the compactness conditions in [28, Theorem 1]. The criterion follows. □

There are two useful consequences. If a compact endpoint x has local exponent $4u$, u odd, at a single prime p, and all $a_q \geq 3$, then a square root of x would have exponent 2 modulo 4 at p and exponent 0 modulo 4 elsewhere. It fails the sum-of-halves condition. More generally, if k_q is odd on a nonempty set S and zero off S, then a square root of $(\prod \theta_q^{k_q})^2/x^{\varepsilon}$ would satisfy

$$\ell_q \equiv k_q - 2\varepsilon u \, \mathbf{1}_{q=p} \pmod{2^{a_q - 1}}. \tag{3.28}$$

Here $\mathbf{1}_{q=p}$ is 1 when $q = p$ and 0 otherwise. Its exponents remain odd on S, contradicting the first condition. Simultaneously inverting the Picard-character convention changes no parity in these assertions.

Lemma 3.12. *Let $A/\mathbf{Q}$ be an abelian variety with good reduction at 2, and let $U \in \mathrm{End}_{\mathbf{Q}}(A)$ have special action $F + V$, with $FV = 2$. Let $n = 2^r \geq 4$, and suppose*

$$B \in A(\mathbf{Q}(\zeta_n))[2^\infty], \qquad \overline{B} = 0, \qquad \sigma B - B = -\frac{\chi_n(\sigma) - 1}{2}\, t,$$

where $t \in A(\overline{\mathbf{Q}})$, $\mathrm{ord}(t) = n/2$, and $Ut = 3t$. The divided coefficient is taken modulo $n/2$ using an odd representative of $\chi_n(\sigma)$. There is a torsion point W with

$$2W = t, \qquad \mathrm{ord}(W) = n, \qquad \sigma W = \chi_n(\sigma) W.$$

Its schematic closure in the good abelian scheme over $\mathbf{Z}_2$ is the finite flat closed subgroup μ_n.

Proof. Factor the characteristic polynomial of U over $\mathbf{Z}_2$ into its coprime unit-root and nonunit-root factors. Their integral Bezout identity gives the Fitting projector $e \in \mathbf{Z}_2[U]$ onto the U-unit part of the two-divisible group. On any finite two-primary group in question it is represented by an integer polynomial in U. It therefore commutes with global Galois and with reduction, and fixes t because $Ut = 3t$. Put $Z = eB$. It is formal and has exactly the same cocycle as B.

On an isocrystal slope strictly between 0 and 1, the commuting operators F and V are both topologically nilpotent, so their sum cannot be invertible. The unit factor is ordinary. The connected part of its two-divisible group is of multiplicative type: its Cartier dual has etale special fiber, hence each finite level of that dual is etale over the complete base. Since an etale torsion

point is determined by its reduction, the formal point Z lies in this connected multiplicative part. Inertia acts on it through the full two-adic cyclotomic character.

Let I_2 be the inertia subgroup at 2 for the chosen embedding. Choose $\tau \in I_2$ with $\chi_{2^\infty}(\tau) = -1$; the two-power cyclotomic tower over $\mathbf{Q}_2$ is totally ramified. The two descriptions of its action give

$$\tau Z = -Z, \qquad \tau Z - Z = t, \qquad \text{hence } 2Z = -t.$$

Set $W = -Z$. Then $2W = t$ and $\mathrm{ord}(W) = n$. The original cocycle now gives, for every global σ,

$$\sigma W = W + \frac{\chi_n(\sigma) - 1}{2} t = \chi_n(\sigma) W.$$

Thus the generated generic subgroup is a global copy of μ_n.

Let $\mathcal{A}$ be the good abelian scheme. The closure of this subgroup inside the finite flat group $\mathcal{A}[n]$ is finite and torsion-free over $\mathbf{Z}_2$, hence finite flat. The group law extends by generic density in flat products. The closure lies in the connected multiplicative factor, so its Cartier dual is a quotient of an etale group scheme and is etale. The displayed global Galois law makes its generic dual the constant group $\mathbf{Z}/n\mathbf{Z}$. A finite etale group over $\mathbf{Z}_2$ with constant generic fiber is constant. Dualizing proves that the closure is μ_n. □

For $A = J_0(M)$ and $U = T_2$, the Eichler–Shimura relation supplies the required special action, and [28, Theorem 6] gives $T_2 = 3$ on $\Sigma(M)$. Theorem 2.4 therefore places the root constructed in Lemma 3.12 inside $\Sigma(M)$ whenever its endpoint is Shimura.

3.8. **The exceptional coset.** Assume every prime factor of N is 1 modulo 8. Choose $p \mid N$, put $a = v_2(p-1)$ and $h = 2^{a-2}$. Csirik's description of the Eisenstein kernel [12, Theorems 5.1(c) and 5.4(b)] gives points x, d whose degeneracy pullbacks to level N satisfy

$$\mathrm{ord}(x) = h, \quad hd = 0, \quad x \in \Sigma(N), \quad \sigma x = \chi_h(\sigma) x, \tag{3.29}$$

$$\sigma d - d = -\frac{\chi_{2h}(\sigma) - 1}{2} x, \qquad I_{\mathrm{tr}} d = 0. \tag{3.30}$$

Both points are defined over $\mathbf{Q}(\zeta_{2h})$. The local character of x is the inflation of a primitive order-h character at p, and thus has local exponent $4u$ for an odd u, in the coordinates of Lemma 3.11. Degeneracy pullback on the Shimura subgroup is character inflation [28, Theorem 4], so the exact order of x is preserved even through repeated prime powers. Good Hecke operators commute with the pullbacks, giving the stated full upper trace relations. Since the field is totally ramified at 2, its reduction $\overline{d}$ is a point of $\mathcal{K}_N$.

Proposition 3.13. *If all prime factors of $N > 1$ are 1 modulo 8, then*

$$\mathcal{K}_N = \overline{C} \ \sqcup \ (\overline{d} + \overline{C}),$$

and the second coset contains no reduction of a rational torsion point of $J_0(N)$. If some prime factor is not 1 modulo 8, then $\mathcal{K}_N = \overline{C}$. In both cases

$$\mathcal{K}_N^\vee \simeq \mathbb{T}/I_{\mathrm{tr}}.$$

Proof. The nonexceptional case follows at once from (3.25). In the exceptional case, suppose a rational torsion point P reduces to $\overline{d}$. Its odd-primary part is in the rational formal kernel and hence is zero by Corollary 2.5. Thus P is two-primary. The point $B = d - P$ is formal and has cocycle (3.30). Apply Lemma 3.12 with $n = 2h$ and endpoint x. It gives a root $2W = x$ with finite flat closure μ_{2h}, so Theorem 2.4 gives $W \in \Sigma(N)$. Its character would have exponent 2 modulo 4 at p and 0 modulo 4 elsewhere, contradicting Lemma 3.11. Therefore $\overline{d}$ has no rational torsion lift.

In particular $\overline{d} \notin \overline{C}$, so (3.25) is attained and the two cosets are exactly the displayed ones. If P reduced to $\overline{d} + \overline{Q}$ with $Q \in C$, the rational torsion point $P - Q$ would reduce to $\overline{d}$, which is

impossible. The local cyclic surjections from $\mathbb{T}/I_{\mathrm{tr}}$ onto $\mathcal{K}_N^\vee$ are now isomorphisms by equality of their orders. This also proves the last assertion in the nonexceptional case. □

Proof of Theorem 3.1. Let $P \in T_N$. At every odd good prime ℓ, reduction is injective on two-primary torsion. Rationality and Eichler–Shimura imply

$$(T_\ell - 1 - \ell)P = 0 \qquad (\ell \nmid 2N).$$

At 2, $\overline{P} \in J(\mathbf{F}_2)$ and $T_2 = F + V$, $FV = 2$, give $(T_2 - 3)\overline{P} = 0$. Hence $\overline{P} \in \mathcal{K}_N$. Proposition 3.13 forces $\overline{P} \in \overline{C}$ in every case. Choose $Q \in C$ with the same reduction. The difference $P - Q$ belongs to the rational formal kernel, which lies in the embedded $\Sigma[2] \subset C$ by Corollary 2.5 and Proposition 3.8. Therefore $P \in C$. The reverse containment follows from Manin–Drinfeld, and the genus-zero levels mentioned at the start complete the assertion. □

4. The Gaussian divisor kernel

Let $M > 1$ be odd with all its prime factors congruent to 1 modulo 4, and write $M = \prod_{q|M} q^{e_q}$. A Gaussian point means an order-two elliptic point on $X_0(M)$. Put $t = \omega(M)$ and

$$G_M = \mathbf{F}_2^t / \langle (1, \dots, 1) \rangle.$$

The set G_M indexes the rational closed Gaussian points, each of degree two. We first justify this description. On a curve with complex multiplication by $\mathbf{Z}[i]$, the i-stable cyclic subgroup of order q^{e_q} is one of the two eigenspaces of i on $E[q^{e_q}]$. The two roots of -1 are distinct because q is odd, and each eigenspace is free of rank one over $\mathbf{Z}/q^{e_q}\mathbf{Z}$. Each is a kernel defined over $\mathbf{Q}(i)$. Conjugation exchanges the choices at every prime simultaneously. The units of $\mathbf{Z}[i]$ preserve every eigenspace and do not identify opposite choices. Since $M > 1$, no orientation is fixed by conjugation. Thus each closed point has exact residue field $\mathbf{Q}(i)$, with the asserted indexing.

For $x \in G_M$ write P_x for this closed point, and define

$$\begin{aligned} \kappa_M : \mathbf{F}_2^{G_M} &\longrightarrow J_0(M)(\mathbf{Q})/(2J_0(M)(\mathbf{Q}) + C_{\mathrm{rat}}(M)), \\ f &\longmapsto \left[\sum_{x \in G_M} f(x)(P_x - 2\infty) \right]. \end{aligned} \tag{4.1}$$

Taking the coefficients as 0 or 1 defines the same element independently of their integral lifts, since the target is killed by 2.

Theorem 4.1. *The kernel of* (4.1) *is the one-dimensional space of constant functions:*

$$\ker \kappa_M = \langle \mathbf{1} \rangle.$$

4.1. **A ramified cover and its half-class.** Let S be a nonempty even-cardinality subset of the primes dividing M. At each $q \in S$ choose a character ϑ_q of $(\mathbf{Z}/q^{e_q}\mathbf{Z})^\times$ of exact order 2^{a_q}, with $a_q = v_2(q-1)$, and put

$$\chi = \prod_{q \in S} \vartheta_q, \qquad r = \max_{q \in S} a_q, \qquad n = 2^r.$$

Each local character sends -1 to -1, so $\chi(-1) = 1$ and χ descends through the diamond group. Its image has exact order n. The intermediate curve X_χ of $X_1(M) \to X_0(M)$ defined by its kernel is a geometrically connected cyclic cover of degree n, defined over $\mathbf{Q}$ with constant deck group.

At a cusp with denominator valuation j at q^{e_q}, the parabolic diagonal is 1 modulo q, because

$$j + \max(e_q - 2j, 0) \geq 1.$$

Thus the character is unramified at every cusp, including all repeated-prime cusps. It kills order-three inertia as well. At the Gaussian orientation ε, let γ_ε be a stabilizer lift. It squares to $-I$, so its image is ± 1. Its value is

$$\chi(\gamma_\varepsilon) = c(-1)^{\sum_{q\in S}\varepsilon_q}, \qquad c = \pm 1. \tag{4.2}$$

The reduced branch divisor D is therefore the divisor of one of the two complementary nonconstant affine patterns associated with S. It is a rational divisor of even degree.

Put $F = \mathbf{Q}(\zeta_n)$ and choose a deck generator g. The cyclic function-field extension has an eigenfunction

$$y \in F(X_\chi)^*, \qquad g^* y = \zeta_n y.$$

This can be obtained by applying a character idempotent to an element of a normal basis; some such projection is nonzero. The function $f = y^n$ is on the base. At a branch point its valuation is $n/2$ modulo n, and elsewhere its valuation is zero modulo n. Hence, for an integral divisor E defined over F,

$$\mathrm{Div}(f) = \frac{n}{2}D + nE, \qquad \deg E = -\deg D/2. \tag{4.3}$$

Define the degree-zero Picard classes

$$H = [-E - (\deg D/2)\infty], \qquad s = [D + 2E], \qquad D_0 = [D - \deg D\,\infty].$$

Then

$$2H = D_0 - s. \tag{4.4}$$

The function y^2 generates the compact unramified quotient of degree $n/2$. This quotient is geometrically connected, and the associated Picard class has exact order $n/2$; otherwise its Kummer equation would split after adjoining constants. Thus

$$s \in \Sigma(M), \qquad \mathrm{ord}(s) = n/2, \qquad \text{the character of } s \text{ is } \chi^2. \tag{4.5}$$

Changing the Picard-character convention simultaneously inverts all these characters and leaves every subsequent parity unchanged.

For global σ, choose an odd representative u of $\chi_n(\sigma)$. Since the cover and its deck generator are rational, σy and y^u have the same eigenvalue. Their quotient h_σ belongs to $F(X_0(M))^*$. Taking divisors in $h_\sigma^n = \sigma f/f^u$ and cancelling n in the free divisor group gives

$$\mathrm{Div}(h_\sigma) = \frac{1-u}{2}D + \sigma E - uE.$$

Substitution in the definitions gives the exact divided cocycle

$$\sigma H - H = -\frac{u-1}{2}s, \qquad \sigma s = us. \tag{4.6}$$

Changing u by n changes the coefficient by $n/2$, which kills s, so the divided coefficient is well-defined. Constants in the Kummer equation and in h_σ have zero divisor.

4.2. **Affine branch divisors.** Suppose that

$$D_0 = 2Q + C_0, \qquad Q \in J_0(M)(\mathbf{Q}), \quad C_0 \in C_{\mathrm{rat}}(M). \tag{4.7}$$

The odd-primary part of C_0 has a half in that same finite rational cuspidal group; absorb it into Q. We may assume C_0 is two-primary. Put $Y = H - Q$. Then

$$2Y = C_0 - s. \tag{4.8}$$

The right side is two-primary torsion, so Y is two-primary torsion.

For an odd good prime $\ell \nmid M$, good reduction is injective on two-primary torsion and gives

$$T_\ell Y = (\sigma_\ell + \ell\sigma_\ell^{-1})Y.$$

Choose odd v with $\ell v \equiv 1$ modulo n. Equation (4.6) gives the exact cancellation

$$(T_\ell - 1 - \ell)Y = -\frac{(\ell-1)+\ell(v-1)}{2}s = -\frac{\ell v - 1}{2}s = 0. \tag{4.9}$$

The last coefficient is divisible by the exact order $n/2$ of s. Since $F/\mathbf{Q}$ is totally ramified at 2, the reduction of Y belongs to $J_0(M)(\mathbf{F}_2)$. On that special fiber $T_2 - 3$ annihilates it. Therefore

$$\overline{Y} \in \mathcal{K}_M.$$

If some prime of M is 5 modulo 8, Proposition 3.13 supplies $P \in C_{\mathrm{rat}}(M)[2^\infty]$ with $\overline{P} = \overline{Y}$; put $\varepsilon = 0$. Otherwise all prime factors are 1 modulo 8. Choose $p \in S$ with $a_p = r$, so that the Csirik parameter at this prime is $h = 2^{r-2} = n/4$. The same proposition gives $P \in C_{\mathrm{rat}}(M)[2^\infty]$ and $\varepsilon \in \{0, 1\}$ such that

$$\overline{Y} = \varepsilon\overline{d} + \overline{P}.$$

Here d, x are defined over $\mathbf{Q}(\zeta_{n/2}) \subset F$. Their cocycle, originally written with $\chi_{n/2}$ before division, can be written with χ_n: changing an odd representative by $j(n/2)$ changes the divided coefficient by $j(n/4)$, which kills x.

Consequently the formal point

$$B = Y - \varepsilon d - P$$

satisfies

$$\sigma B - B = -\frac{\chi_n(\sigma) - 1}{2}(s - \varepsilon x).$$

The endpoint $s - \varepsilon x$ has exact order $n/2$. If $\varepsilon = 1$, this follows because s has that order whereas x has the strictly smaller order $n/4$. Both are Shimura, so $T_2(s - \varepsilon x) = 3(s - \varepsilon x)$. Lemma 3.12 and Theorem 2.4 give a compact Shimura character for a point W with $2W = s - \varepsilon x$.

In the coordinates of Lemma 3.11, write the character χ as $\prod \theta_q^{k_q}$. The exponent k_q is odd on S and zero elsewhere. The character of s has exponent $2k_q$; the optional correction x has exponent $4u$, u odd, at the single prime p. The character of W must therefore satisfy (3.28). Its exponent is odd at every $q \in S$, contrary to Lemma 3.11. This proves that (4.7) is impossible for the branch pattern of (4.2).

4.3. **Compression to affine patterns.**

Proof of Theorem 4.1. The complete Gaussian divisor gives the constant relation. Indeed, in $\mathrm{Div}(j - 1728)$ the order-two elliptic points have multiplicity one, all other interior zeros have even multiplicity, and the polar divisor is rational and cuspidal. Normalize each term to degree zero at ∞. Its class shows

$$\mathbf{1} \in \ker \kappa_M. \tag{4.10}$$

This proves the theorem if $t = 1$, because the domain then has a single coordinate.

Proceed by induction on t. For $t \geq 2$, forget an entire prime power q^{e_q}. The lower level still has an odd split prime, so the upper and lower Gaussian closed points both have exact residue field $\mathbf{Q}(i)$. Their divisor pushforward coefficient is

$$[\mathbf{Q}(i) : \mathbf{Q}(i)] = 1.$$

The cusp ∞ also pushes forward with coefficient one. Thus a normalized Gaussian pair class pushes to the normalized lower pair class. This pushforward preserves rational doubles and rational cusp divisors.

Let $\mathbf{e}_q$ be the class in G_M of the standard basis vector indexed by q. On the orientation quotient the fibers are $\{x, x + \mathbf{e}_q\}$. If $f \in \ker \kappa_M$, its pushforward is

$$P_q f(x) = f(x) + f(x + \mathbf{e}_q) \in \ker \kappa_{M/q^{e_q}}.$$

Induction says that every $P_q f$ is constant. Thus there are $c_q \in \mathbf{F}_2$ with $f(x+\mathbf{e}_q) - f(x) = c_q$. The flips $\mathbf{e}_q$ generate G_M and have the sole relation $\sum_q \mathbf{e}_q = 0$. Following that relation gives $\sum_q c_q = 0$. The c_q therefore define a linear functional λ on G_M, and successive flips yield

$$f(x) = f(0) + \lambda(x).$$

If $\lambda \neq 0$, its coefficients define a nonempty even subset S. Equation (4.2) realizes either this affine pattern or its complement as the branch divisor of the cyclic cover just constructed. By (4.10), both have the same kernel status. The preceding argument excludes such a pattern. Hence $\lambda = 0$ and f is constant.

□

5. The semistable two-primary case

Throughout the dyadic arguments below, M is odd, and we write

$$T_L = J_L(\mathbf{Q})[2^\infty], \qquad C_L = C_{\mathrm{rat}}(L)[2^\infty].$$

Theorem 5.1. *For every odd positive integer M, one has $T_{2M} = C_{2M}$.*

Let $U = U_2$ act in the Picard convention. For a finite two-primary group H preserved by U, choose $r \geq 1$ such that the kernels and images of its powers have stabilized. The Fitting decomposition is

$$H = H^{\mathrm{nil}} \oplus H^{\mathrm{ord}}, \qquad H^{\mathrm{nil}} = \ker U^r, \quad H^{\mathrm{ord}} = \operatorname{im} U^r.$$

Indeed, if $x = U^r y$ and $U^r x = 0$, then stabilization gives $y \in \ker U^{2r} = \ker U^r$, so $x = 0$; the cardinality formula for U^r gives the direct sum. The operator U is nilpotent on the first summand and invertible on the second. If $U^m = 1$ on H^{ord}, then U^{rm} is the projection $e_H : H \to H^{\mathrm{ord}}$. These projections are functorial and preserve $C_{2M} \subset T_{2M}$, because pullback and norm preserve rational cuspidal divisors. On a finite free $\mathbf{Z}_2$-module with a U-action, the compatible projections modulo 2^a define the ordinary projector $e = \lim_{r\to\infty} U^{r!}$.

5.1. **The nilpotent factor.** Put $A = J_M$, $a = T_2 \in \mathrm{End}(A)$, and use the two degeneracies to define

$$\delta = (\alpha^*, \beta^*) : A^2 \longrightarrow J_{2M}, \qquad \delta' = (\alpha_*, \beta_*) : J_{2M} \longrightarrow A^2.$$

In the Picard convention fixed above,

$$U\delta = \delta V, \quad V = \begin{pmatrix} a & 2 \\ -1 & 0 \end{pmatrix}, \qquad \delta'\delta = \begin{pmatrix} 3 & a \\ a & 3 \end{pmatrix}.$$

Let $B = \left(\begin{smallmatrix} -1 & a \\ 0 & -1 \end{smallmatrix}\right)$. Multiplication gives $B\delta'\delta = V^2 - 1$. In fact

$$\delta B \delta' = U^2 - 1 \quad \text{on } J_{2M}. \tag{5.1}$$

It holds on the old image by the matrix identity. On the connected new subvariety, $\delta' = 0$ and $U = -w_2$, so it holds there too. The old and new images generate the Jacobian up to isogeny; equality on these images is equality of homomorphisms. Theorem 3.1 makes $\delta' P$ cuspidal for $P \in T_{2M}$. Thus $(U^2-1)P \in C_{2M}$. If $U^r P = 0$ and $2s \geq r$, then

$$P = -\sum_{j=0}^{s-1} U^{2j}(U^2-1)P \in C_{2M}.$$

This proves the theorem on the entire nilpotent factor.

5.2. **An integral ordinary Frobenius quotient.**

Lemma 5.2. *For $P \in J_{2M}(\mathbf{Q}_2)[2^\infty]^{\mathrm{ord}}$, the point $(U-1)P$ reduces to the identity point of the original Néron model. Consequently $UP = P$ for $P \in T_{2M}^{\mathrm{ord}}$.*

Proof. Let $k = \mathbf{F}_2$, and let $\mathcal{J}/\mathbf{Z}_2$ be the semistable Néron model. In the quasi-finite flat group $\mathcal{J}^0[2^n]$, take the finite part whose special fiber is $\mathcal{J}_k^0[2^n]$. These form the finite-identity two-divisible group $\mathscr{H}$. Its connected part $\mathscr{C}$ is finite flat at each finite level and specializes, on geometric points, only to the identity. This construction is functorial; it is the usual finite-part construction in semistable reduction, including at the residue characteristic [21, IX, §5][10, §5].

Set $T = T_2 J_{2M}$ and $T_c = T_2\mathscr{C}$. The inclusions

$$T_c/2^n T_c = \mathscr{C}[2^n](\overline{\mathbf{Q}}_2) \hookrightarrow J_{2M}[2^n](\overline{\mathbf{Q}}_2) = T/2^n T$$

show $T_c \cap 2^n T = 2^n T_c$. Hence T/T_c is free. Let e be the U-unit idempotent on T/T_c. The free quotient $E = e(T/T_c)$ occurs in the exact sequence

$$0 \longrightarrow \mathscr{C}^{\mathrm{ord}}(\overline{\mathbf{Q}}_2) \longrightarrow J_{2M}[2^\infty](\overline{\mathbf{Q}}_2)^{\mathrm{ord}} \longrightarrow E \otimes_{\mathbf{Z}_2} (\mathbf{Q}_2/\mathbf{Z}_2) \longrightarrow 0. \tag{5.2}$$

We claim that inertia is trivial on E and arithmetic Frobenius is U. It suffices to check both assertions on $E \otimes \mathbf{Q}_2$, since E is a free integral quotient.

For the good old factor J_M, let L be the étale Tate module of its special two-divisible group. It is unramified, and its Frobenius F is invertible. Eichler–Shimura gives $a = F + 2F^{-1}$. The matrix

$$S = \begin{pmatrix} -F & -2F^{-1} \\ 1 & 1 \end{pmatrix} \quad \text{satisfies} \quad S^{-1}\begin{pmatrix} a & 2 \\ -1 & 0 \end{pmatrix} S = \begin{pmatrix} F & 0 \\ 0 & 2F^{-1} \end{pmatrix}.$$

Its determinant $2F^{-1} - F = -F^{-1}(F^2 - 2)$ is a unit over $\mathbf{Z}_2$. Therefore the ordinary factor of L^2 has $U = F$ integrally.

The two-new factor is purely toric: the abelian part of the special identity of J_{2M} is the product of the two copies of J_M coming from the two main components, and the old map is an isogeny onto that part. For a purely toric semistable abelian variety, Raynaud uniformization gives

$$0 \longrightarrow \mathscr{T}[2^n] \longrightarrow B[2^n] \longrightarrow Y/2^n Y \longrightarrow 0,$$

where Y is the unramified period lattice, the character lattice of the dual Raynaud torus. The toric two-divisible group is connected, so the quotient by T_c retains exactly $Y \otimes \mathbf{Z}_2$. In particular it retains the directions responsible for nonidentity components.

Orient the edges in the modular reduction graph from the first main component to the second. The involution w_2 exchanges these components and applies Frobenius to the enhanced supersingular label, since the unique subgroup of order two at a supersingular point is $\ker F$. Thus $-w_2$ acts as Frobenius on H_1 of the graph. Resolving a thick node subdivides an edge and leaves this assertion unchanged. The two identities

$$U + w_2 = \alpha^*\beta_*, \qquad U^\dagger + w_2 = \beta^*\alpha_*$$

show that both U and its Rosati adjoint act as $-w_2$ on the torus: the right hand sides factor through a good abelian variety, and a homomorphism from a torus to an abelian variety is zero. Hence the character and period conventions give the same Frobenius action [43, Proposition 3.8]. The rational old/new isogeny now proves the claim on $E \otimes \mathbf{Q}_2$, and thus on E.

The image of P in the last term of (5.2) is Galois fixed. Therefore $(U-1)P$ lies in $\mathscr{C}^{\mathrm{ord}}$; properness of its finite flat level makes it reduce to the identity. For globally rational P, Corollary 2.5 puts this difference in $\Sigma(2M)[2]$. Ling–Oesterlé's Hecke formula gives $U = [2]$ there, for both correspondence directions [28, Theorem 6]. The difference is consequently both U-nilpotent and U-ordinary, and is zero. □

Let $\mathbb{T}^{\mathrm{ord}}$ denote the complete ordinary cuspidal Hecke order, and put

$$I = (U - 1,\ T_\ell - 1 - \ell : \ell \nmid 2M). \tag{5.3}$$

The lemma and good-prime Eichler–Shimura prove that I annihilates T_{2M}^{ord} at every two-power order.

5.3. **Logarithmic differentiation.** The socle estimate requires a map on all rational two-torsion, including points on nonidentity components. We give its construction and its compatibility with correspondences.

Lemma 5.3. *Let $\mathscr{Y}/\mathbf{Z}_2$ be a regular arithmetic surface with smooth generic curve. If a rational function g has only vertical divisor on an open subset, then $d\log g$ belongs to the relative dualizing sheaf on that subset.*

Proof. The dualizing sheaf is invertible, so membership may be tested at codimension one. At a horizontal point g is a unit and $\Omega^1 \to \omega$ gives the assertion. At a vertical generic DVR B_0, choose a separating parameter $\bar{z}$ for its residue function field over $\mathbf{F}_2$, and lift it to z. The restricted valuation on $\mathbf{Q}_2(z)$ is the Gauss valuation, because a polynomial with a unit coefficient has nonzero reduction at the transcendental $\bar{z}$. Thus its valuation ring is $A_0 = \mathbf{Z}_2[z]_{(2)}$. Normalization in the finite function-field extension and localization recover B_0. On completion there is a finite extension of complete DVRs $A = \widehat{A}_0 \to B = \widehat{B}_0$.

Lift the finite separable residue extension by Hensel's lemma to a finite étale A'/A inside B. Then B/A' is totally ramified of degree $e = v_B(2)$. A uniformizer t has degree at least e over $\mathrm{Frac}(A')$ by its valuation, hence has degree exactly e. Its minimal polynomial is Eisenstein,

$$F(T) = T^e + \sum_{i<e} a_i T^i, \qquad a_i \in 2A', \quad a_0 \notin 4A',$$

and $B = A'[t]$, since the latter is already a complete DVR with the same fraction field. Finite duality and the monogenic different identify the completion of the *geometric* dualizing lattice as

$$\widehat{\omega}_{\mathscr{Y}/\mathbf{Z}_2, B_0} = F_T(t)^{-1} B\, dz. \tag{5.4}$$

Indeed one may first normalize an open of $\mathbf{P}^1_{\mathbf{Z}_2}$ in the function field and compute its complementary module at B_0; completion preserves this finite duality, and the étale factor has unit different.

The derivation ∂_z preserves A, and extends integrally over the étale algebra A'. Thus $\partial_z a_i \in 2A'$. Differentiating $F(t) = 0$ gives

$$\frac{dt}{t} = -\frac{\sum_{i<e} (\partial_z a_i) t^i / t}{F_T(t)}\, dz.$$

The numerator before division by t is in $2B = t^e B$, so the displayed numerator is in B. Equation (5.4) proves integrality. Completion is faithfully flat, so it descends to B_0. Writing $g = t^n u$ with $u \in B_0^*$ completes the proof, because du/u is integral and $n\, dt/t$ is integral. □

For a smooth proper geometrically connected curve $C/\mathbf{Q}_2$ with a rational point, let Λ_C be its canonical integral dualizing lattice. An admissible model in the sense of [3] is a proper flat normal model that is cohomologically flat in dimension zero and has rational singularities. A proper regular model with a section is admissible [3, Proposition 3.3]. If two smooth proper curves admitting admissible models are related by a finite generically étale map, there are admissible models over the *same* discrete valuation ring on which the map is finite [3, Theorem 3.4]. Every modular curve used here has the rational cusp at infinity, and every finite map of our characteristic-zero curves is generically étale. These are precisely the hypotheses needed, including for source $X_0(4M)$. Theorem 4.4 and Propositions 4.5–4.6 of [3] identify the global dualizing lattices and give integral pullback and trace. Different degeneracy maps may use different finite normal models; canonical model independence permits their composition.

Proposition 5.4. *There is a canonical additive map*

$$\theta_C : J_C(\mathbf{Q}_2)[2] \longrightarrow \Lambda_C/2\Lambda_C$$

which intertwines Picard pullback and norm with differential pullback and trace. If C has a regular semistable model, then

$$\ker \theta_C = J_C^1(\mathbf{Q}_2)[2].$$

Here $J_C^1(\mathbf{Q}_2)$ is the kernel of reduction to the identity point of the Néron model.

Proof. Represent P by a rational degree-zero divisor D, and choose f with $\mathrm{Div}_C(f) = 2D$. The rational point on C ensures the existence of this divisor representative. On a regular model the horizontal closure $\bar{D}$ is Cartier. If h is a local Cartier equation, then f/h^2 has vertical divisor and

$$\eta_h = d\log f - 2d\log h \in \omega$$

by Lemma 5.3. On overlaps the difference is in 2ω. These reductions define a section of $\omega/2$. Changing D changes its horizontal equations as well as the square factor in f; the remaining vertical factor again has integral dlog. Thus the class is independent of choices. Products prove additivity. Torsion-freeness of $H^1(\omega)$ gives $H^0(\omega)/2 = H^0(\omega/2)$.

On a regular semistable model write

$$\mathrm{Div}(f) = 2\bar{D} + \sum_i n_i C_i.$$

At a regular node $uv = 2$, the local expression $f = h^2 u^a v^b \epsilon$ gives

$$\eta_h = (a-b)\frac{du}{u} + d\log\epsilon. \tag{5.5}$$

Its residue on the edge $C_i - C_j$ is $n_i - n_j \pmod 2$. Every exceptional component resolving a thick node is retained here. If $\theta_C(P) = 0$, connectedness makes all n_i have the same parity. Since the special fiber is reduced, division of f by a power of two makes its whole divisor $2E$ for an integral Cartier divisor E. The line bundle $L = \mathcal{O}(E)$ is two-torsion on the model and has degree zero on every component. On a smooth normalized component, zero Kummer differential says $dg = 0$ for its square-trivializing function. Over a perfect field of characteristic two this means $g = h^2$; the torsion-free divisor group makes the restricted line bundle trivial. The kernel of normalization on the multidegree-zero Picard group of a nodal curve is a torus. Its geometric points have no two-torsion in characteristic two. Thus L_k is trivial. The regular semistable Picard/Néron comparison places P in J_C^1.

Conversely, a formal two-torsion point is represented in Pic^0 by L with L_k trivial. Its square is $\mathcal{O}(V)$ for a vertical divisor whose intersection with every component is zero. The intersection form is the negative sum of the squares of differences across edges; connectedness makes V a multiple of the whole reduced fiber. This divisor is principal, so L^2 is trivial on the model. Relative to the trivialization of L_k, its square trivialization is a global special unit, hence a constant. Its dlog is zero. This proves the kernel claim.

For naturality, take normal finite admissible models over the same $\mathbf{Z}_2$ for a finite map $h : C' \to C$; their existence is part of Cais's Theorem 3.4. Transfer the construction from regular resolutions by the canonical birational trace. At a vertical generic DVR B_0, an integral lift of $\theta(P)$ is characterized by

$$\widetilde{\theta}(P) \equiv d\log f \pmod{2\omega_{B_0}}. \tag{5.6}$$

This congruence is on the full Artin ring $B_0/2B_0$, including its nilpotents. The relative canonical module of the special fiber has no embedded associated primes: the normal arithmetic surface is Cohen–Macaulay, its dualizing module is maximal Cohen–Macaulay, and two is a nonzerodivisor. Consequently these complete generic-stalk congruences determine the global class.

Integral differential pullback preserves (5.6). For norm one uses the exact characteristic-zero identity

$$d\log \mathrm{Nm}(g) = \mathrm{Tr}_h(d\log g).$$

Over a target vertical generic point the source is a finite semilocal normalization. The congruence at each of its maximal DVRs holds in its finite dualizing module, and the integral dualizing trace preserves 2ω. Purity gives both equalities globally. Model independence allows different finite models for the two sides of a correspondence. □

For example, if $t^2 = 2z$ and $\eta = dz/(2t)$ generates the dualizing lattice, then

$$d\log t = (t/z)\eta, \qquad \mathrm{Tr}(\eta) = 0, \qquad \mathrm{Tr}(t\eta) = dz.$$

The coefficient t/z vanishes on the reduced component but not modulo two. Moreover $\mathrm{Tr}(d\log t) = dz/z = d\log(-2z)$, the logarithmic differential of the norm.

For $U_2 = \beta_*\alpha^*$ on $X_0(2M)$ the differential operator is $\mathrm{Tr}_\beta\,\alpha^*$. At the cusp, $\alpha^* q = t$ and $\beta^* q = t^2$, whence in characteristic zero

$$\mathrm{Tr}_\beta\,\alpha^*\left(f(q)\frac{dq}{q}\right) = \frac{f(q^{1/2}) + f(-q^{1/2})}{2}\frac{dq}{q} = \sum_n a_{2n}(f)q^n\frac{dq}{q}.$$

This identifies the operator with Fourier U_2; Proposition 5.4 gives its integrality. The same calculation gives the bad odd operators, and the extra good-prime branch gives the usual T_ℓ. Thus equivariance is for the full Hecke algebra with the fixed Picard convention.

5.4. **The ordinary socle.** Let $\Lambda = H^0(\mathscr{X}, \omega)$ for the regular semistable model of $X_0(2M)$, and let S_q be the lattice of cusp forms with integral expansion at ∞. Edixhoven's exact comparison for $2 \parallel N$ gives, for $Q = S_q/\Lambda$,

$$4Q = 0, \qquad UQ[2] = 0$$

[16, Propositions 2.7 and 2.9]. Hence $U^2Q = 0$: $2Ux = 0$ first, and then $U(Ux) = 0$. It follows that

$$\Lambda^{\mathrm{ord}} = S_q^{\mathrm{ord}}. \tag{5.7}$$

The comparison applies to every odd M.

The pairing $(t, f) \mapsto a_1(tf)$ identifies

$$S_q^{\mathrm{ord}} = \mathrm{Hom}_{\mathbf{Z}_2}(\mathbb{T}^{\mathrm{ord}}, \mathbf{Z}_2).$$

Indeed $a_1(T_n f) = a_n(f)$ separates forms. If $a_1(tf) = 0$ for all f, commutation with T_n makes every coefficient of tf vanish, so $t = 0$. The rational pairing is thus perfect on both sides; the integral Hecke algebra is spanned by the T_n, so its dual is exactly the coefficient lattice. Proposition 5.4 and Corollary 2.5 give a Hecke-equivariant injection

$$T_{2M}^{\mathrm{ord}}[2] \hookrightarrow (\Lambda/2\Lambda)^{\mathrm{ord}}:$$

an ordinary point in its kernel is formal, hence in $\Sigma(2M)[2]$, where U kills it, so it is zero. If $A = \mathbb{T}^{\mathrm{ord}}/2$ and $\mathfrak{m}$ is maximal, then

$$\mathrm{Hom}_{\mathbf{F}_2}(A, \mathbf{F}_2)[\mathfrak{m}] = \mathrm{Hom}_{\mathbf{F}_2}(A/\mathfrak{m}, \mathbf{F}_2)$$

has dimension one over $A/\mathfrak{m}$. Consequently

$$\dim_{\mathbb{T}^{\mathrm{ord}}/\mathfrak{m}} T_{2M}^{\mathrm{ord}}[\mathfrak{m}] \le 1. \tag{5.8}$$

5.5. **The ordinary Eisenstein index.** Put $R = \mathbf{Z}_2$ and $K = \mathbf{Q}_2$. Let S_K be the entire ordinary cusp space and set

$$F_d = \frac{d}{24}(E_2(dz) - 2E_2(2dz)), \quad d \mid M, \qquad E_K = \sum_{d|M} KF_d, \quad V = S_K \oplus E_K.$$

For $f_a = \sum a_d F_d$, the positive coefficient at n equals

$$-\sum_{d|\gcd(n,M)} da_d\, \sigma_1(\mathrm{oddpart}(n/d)). \tag{5.9}$$

At indices $n, d \mid M$ the matrix is triangular with unit diagonal $-d$. Thus its positive-integral and fully integral lattices are

$$E^+ = \{f_a : a \in R^{\tau(M)}\}, \qquad E = \{f_a : a \in R^{\tau(M)},\ \textstyle\sum da_d \in 8R\}. \tag{5.10}$$

The constant of f_a is $-\sum da_d/24$, and $E^+/E = R/8R$.

Let $\widetilde{\mathbb{T}}$ be the integral ordinary Hecke image on V, and $\mathbb{T}$ its image on S_K. The coefficient argument just used, on both sides of the pairing $a_1(tf)$, gives

$$M^+ = \{f \in V : a_n(f) \in R\ (n > 0)\} = \mathrm{Hom}_R(\widetilde{\mathbb{T}}, R), \qquad S = S_K \cap R[[q]] = \mathrm{Hom}_R(\mathbb{T}, R).$$

Both Hecke orders are finite free: their stable cusp coefficient lattice and explicit Eisenstein lattice provide integral realizations. Reduction of M^+ into positive series is injective, since divisibility of every positive coefficient by two means $f/2 \in M^+$.

Lemma 5.5. *In $\widetilde{\mathbb{T}}$ the ideal $I_{\mathrm{tr}} = (U - 1, T_\ell - 1 - \ell : \ell \nmid 2M)$ equals $I_E = \mathrm{Ann}(E_K)$.*

Proof. The inclusion $I_{\mathrm{tr}} \subseteq I_E$ is immediate. For an odd $q^e \parallel M$, continuity and Chebotarev applied simultaneously to the finitely many newform Galois representations give a scalar trace operator $t(\sigma)$ in the complete Hecke order with $t(\sigma) \equiv 1 + \chi_{2^\infty}(\sigma) \pmod{I_{\mathrm{tr}}}$. Choose a lift of arithmetic Frobenius at the odd prime q. Its cyclotomic value is q. The full old blocks have annihilating polynomials

conductor exponent	polynomial	trace
0	$X^{e-1}(X^2 - a_q X + q)$	a_q
1	$X^{e-1}(X - a_q)$	$a_q(1+q), \quad a_q = \pm 1$
≥ 2	X^{e-1}	not needed.

They imply, on every block and on the selected Eisenstein space, $U_q^{e+1} + qU_q^{e-1} = U_q^e t(\sigma)$. In the conductor-one case the remaining evaluation is $q(1 - a_q^2) = 0$; in higher conductor every term already contains the annihilating nilpotent factor. Therefore

$$U_q^{e-1}(U_q - 1)(U_q - q) \in I_{\mathrm{tr}}. \tag{5.11}$$

A positive series modulo two killed by the ideal is determined by its coefficients at $d \mid M$. The U_2 relation removes powers of two, the good-prime Hecke recurrence removes good primes, and (5.11) limits each bad exponent to $0, \ldots, e$. These recurrences commute. Thus the dual of $(\widetilde{\mathbb{T}}/I_{\mathrm{tr}})/2$ has dimension at most $\tau(M)$. On the other hand the quotient surjects onto $\widetilde{\mathbb{T}}/I_E$, which is free of rank $\tau(M)$ by coefficient duality on E_K. This surjection splits as an R-module map. A nonzero kernel would, by Nakayama, add a generator modulo two. The kernel is therefore zero. □

For $d \mid M$ put $g_d = \gcd(d, M/d)$ and $w_d = \varphi(g_d)$. The two rational cusp orbits of denominators $d, 2d$ both have degree w_d. Define the invertible matrix

$$A_M(d, e) = \frac{M \gcd(d, e)^2}{g_d de}. \tag{5.12}$$

The full cusp-order matrix at $2M$ is $A_M \otimes \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$. For the exponent pattern $r_d = a_d, r_{2d} = -a_d$, its residues are $(b, -b)$ with $b = A_M a/24$. The weighted column sums are

$$(w_d)_d^t A_M = \psi(M)(1, \dots, 1). \tag{5.13}$$

To verify this for q^e, split the row sum at the column index j. The endpoint factors $\varphi(q^{\min(i,e-i)})/q^{\min(i,e-i)}$ are one, the interior factors are $1 - 1/q$, and the two geometric sums give $q^{e-1}(q+1)$ independently of j. Tensoring proves the formula for all M, including repeated prime factors.

On the two-part eta exponent block the operator is $\left(\begin{smallmatrix} 3 & 2 \\ -1 & 0 \end{smallmatrix}\right)$. Conjugation by $\left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$ gives on residues

$$U_{\mathrm{res}} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}, \qquad e_{\mathrm{res}} = 2 - U_{\mathrm{res}} = \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}.$$

Hence the ordinary part of the weighted degree-zero rational divisor lattice is exactly

$$D^{\mathrm{ord}} = \{(b, -b) : b \in R^{\tau(M)}\}. \tag{5.14}$$

The idempotent e_{res} preserves the integral residue lattice.

The completed lattice of rational unit periods in E_K is exactly E. Invert the rational cusp-order matrix on the divisor of a rational unit. The weighted column identity ensures weight zero. Clearing rational exponents and multiplying to satisfy the eta congruences and character condition produces an eta quotient with the same divisor as a power of the unit. Their ratio is constant, so the unit's dlog belongs to the rational eta span. At the rational cusp, regularity and disjoint cusp closures give $u = 2^v q^a \epsilon(q)$ with $\epsilon \in R[[q]]^*$. Its dlog is integral. Equation (5.9) and the constant therefore put its ordinary part in E.

Conversely, take an integral vector a with $\sum d a_d \in 8\mathbf{Z}$. For $r_d = a_d, r_{2d} = -a_d$, the two eta congruence sums are $A = -\sum d a_d$ and $B = \sum (M/d) a_d$. Odd squares are one modulo eight, so $B \equiv -MA \pmod 8$. Every odd-prime character moment cancels, and the two-character moment $-\sum a_d$ is even. At any prescribed two-adic precision, an odd multiplier congruent to one at that precision and divisible by three makes A, B divisible by 24. Newman's criterion [47, Theorem 1.1] then gives a rational modular unit. Approximation proves the claimed equality of completed period lattices.

5.6. **The logarithmic lattice.** On the regular semistable model retaining every cusp, let C be the disjoint reduced cusp closure and put $L = H^0(\omega(C))$. The residue sequence is

$$0 \longrightarrow H^0(\omega) \longrightarrow L \xrightarrow{\mathrm{Res}} \ker\bigl(\mathrm{Tr} : H^0(C, \mathcal{O}_C) \to R\bigr) \longrightarrow 0. \tag{5.15}$$

Its final surjectivity is integral: on every connected reduced fiber, $H^0(\mathcal{O}(-C)) = 0$, so duality gives $H^1(\omega(C)) = 0$; cohomology and Nakayama give the same vanishing over R. The connecting map into $H^1(\omega) = R$ is the sum of residue-field traces. We insert the explicit orbitwise rational vectors with weights w_d into this kernel.

The lattice L is stable under both differential directions of Hecke correspondences. To justify this integrally, use the generalized Jacobian with reduced cusp modulus. The smooth rigidified Picard functor is first pushed out along its identity torus into the Néron model of that torus; this preserves identity cotangents. The quotient by the étale closure of zero is the generalized Néron model. Its cotangent is therefore $H^1(\mathcal{O}(-C))^\vee = L$. Pullback and norm for finite maps preserving the cusp sets extend to these Néron models [23, §§1.3, 1.6, 1.8 and Theorem 1.16].

Let $M_0 \subset V$ be the lattice of forms with all coefficients, including the constant, in R. Then

$$M_0 = L \cap V, \qquad \mathrm{Res}(M_0) = D^{\mathrm{ord}}, \qquad M_0/(S+E) \simeq C_{2M}^{\mathrm{ord}}. \tag{5.16}$$

For the nontrivial inclusion, a fully integral cusp germ has nonnegative valuation on the component through ∞. Its residues at every cusp on that component are integral. The anti-diagonal pattern makes the residues on the opposite component integral as well. Lift this vector by (5.15), and

apply the integral ordinary projector. If its vector is $(b,-b)$, the unique Eisenstein representative is F_a with $a = 24A_M^{-1}b$. Thus the projected lift minus F_a is ordinary cuspidal, so the lift belongs to V. Subtracting it from the given form leaves a q-integral ordinary cusp form; it belongs to $H^0(\omega)$ by (5.7). This proves the first two assertions. Quotienting the divisor sequence by the completed unit periods E proves the last one. Principal rational cusp divisors are represented by rational functions: the Galois ratios of a function are constants and Hilbert 90 rescales it.

We next prove that every element of M^+ has constant denominator at most eight. If its constant has exact denominator 2^b with $b > 0$, scale it to a form

$$H(q) = 1 + 2^b \sum_{n>0} c_n q^n, \qquad c_n \in R.$$

Choose an odd full auxiliary level L_0 divisible by $3M$, and an unramified coefficient ring W containing the required roots of unity. On the proper fine Deligne–Rapoport curve with this level and a cyclic order-two subgroup, a fixed $2^r H$ extends as an integral Hodge-square section: localization of coherent sections on the proper model clears one denominator. Pull it to the formal ordinary locus of the tame curve by choosing its canonical connected subgroup of order two. If $r > 0$, reduction of this section vanishes at the chosen cusp, hence on the geometrically integral ordinary curve. It is globally divisible by two there. Iterating proves that H itself is integral on that formal ordinary locus.

Katz's finite Igusa covers exist over every $W/2^m$ for our odd auxiliary level, have connected special fiber, and have an injective one-cusp q-expansion map with flat cokernel [24, §1.0, §1.2.3, and Lemma 1.4.3]. Let $T_{b,b}$ be the Igusa cover over $W/2^b$ parametrizing trivializations of the connected 2^b-torsion by μ_{2^b}. Its canonical differential ω_{can} pulls back to dT/T under the universal trivialization, where T is the multiplicative coordinate on μ_{2^b}. Put $h = H/\omega_{\mathrm{can}}^2$. Its expansion is one on the full $W/2^b$-thickening, so $h = 1$ there. The deck transformation $3 \in (\mathbf{Z}/2^b)^*$ changes ω_{can} to $3^{-1}\omega_{\mathrm{can}}$, while fixing the underlying canonical ordinary curve. It sends h to $9h$. Therefore $1 = 9$ in $W/2^b$, and $b \le v_2(8) = 3$. The form F_1 has constant $-1/24$, so the bound is attained. Since $E^+/E = R/8R$ supplies all these constants,

$$M^+ = M_0 + E^+. \tag{5.17}$$

The submodule $E^+ = M^+ \cap E_K$ is saturated. Its quotient is free, and its dual is I_E under the coefficient pairing. Apply integral duality to

$$0 \longrightarrow S \longrightarrow M^+/E^+ \longrightarrow X^+ \longrightarrow 0, \qquad X^+ = M^+/(S + E^+).$$

Using $\mathrm{Ext}^1_R(X^+, R) = \mathrm{Hom}(X^+, K/R)$ gives

$$0 \longrightarrow I_E \longrightarrow \mathbb{T} \longrightarrow (X^+)^\vee \longrightarrow 0.$$

Lemma 5.5 identifies the last quotient with $\mathbb{T}/I$. Equations (5.16)–(5.17), and $M_0 \cap (S + E^+) = S + E$, identify X^+ with C^{ord}_{2M}. Thus

$$\mathbb{T}^{\mathrm{ord}}/I \simeq (C^{\mathrm{ord}}_{2M})^\vee. \tag{5.18}$$

5.7. **The ordinary torsion group.** Let $A = T^{\mathrm{ord}}_{2M}$. Finite duality identifies the dual of $A^\vee/\mathfrak{m}A^\vee$ with $A[\mathfrak{m}]$. Equation (5.8) and Nakayama make each local factor of $A^\vee$ at most cyclic over the local Hecke order. Since $IA = 0$ by Lemma 5.2, there is a surjection from each local factor of $\mathbb{T}^{\mathrm{ord}}/I$ to the corresponding factor of $A^\vee$. The full index (5.18) gives $|A| \le |C^{\mathrm{ord}}_{2M}|$. The reverse inclusion is a subgroup inclusion, so equality holds. With (5.1) this proves Theorem 5.1.

The ordinary cusp group also has an explicit abstract description. Modulo eight, every row of A_M is a unit multiple of $(e)_{e|M}$, since $A_M(d,e) \equiv M g_d d\, e \pmod 8$. Therefore $E = \{a : A_M a \in 8R^{\tau(M)}\}$. The map $b \mapsto 24b \bmod A_M R^{\tau(M)}$ has kernel $A_M E/24$ and image $8\,\mathrm{coker}_R A_M$. Hence

$$C^{\mathrm{ord}}_{2M} \simeq 8\,\mathrm{coker}_R A_M. \tag{5.19}$$

For $q^e \parallel M$, discard the unit row and column factors in A_{q^e} to obtain $G_{ij} = q^{2\min(i,j)}$, $0 \le i, j \le e$. If L_e has diagonal one and subdiagonal -1, consecutive differences give the exact identity

$$L_e G L_e^t = \operatorname{diag}(1, q^2-1, q^2(q^2-1), \dots, q^{2e-2}(q^2-1)).$$

Off-diagonal entries cancel and the positive diagonal entries are $q^{2i} - q^{2i-2}$. Since $\det L_e = 1$, the local elementary divisors are one and e copies of 2^{t_q}, where $t_q = v_2(q^2-1)$. Tensoring the prime-power matrices and multiplying the cokernel by eight proves that, for each nonempty subset S of the prime divisors of M, there is a cyclic factor of exponent $\sum_{q\in S} t_q - 3$, with multiplicity $\prod_{q\in S} v_q(M)$. Zero exponents are omitted. In particular

$$\operatorname{length}_R C_{2M}^{\mathrm{ord}} = \sum_{q^e \parallel M} \frac{\tau(M) e}{e+1} v_2(q^2-1) - 3\tau(M) + 3.$$

6. The level $4M$

Theorem 6.1. *For every odd positive integer M,*

$$J_0(4M)(\mathbf{Q})[2^\infty] = C_{\mathrm{rat}}(4M)[2^\infty].$$

More precisely, the three transposition quotients of the full-two cover induce a surjection

$$(T_{2M}/C_{2M})^3 \twoheadrightarrow T_{4M}/C_{4M}.$$

6.1. **The symmetric-group diagram.** Conjugation by $\tau \mapsto 2\tau$ identifies $X = X_0(4M)$ with the curve for $\Gamma(2) \cap \Gamma_0(M)$. Reduction modulo two gives its $\mathbf{Q}$-defined deck group $\mathrm{SL}_2(\mathbf{F}_2) = S_3$ over $B = X_0(M)$. Let c generate A_3, put $Z = X/\langle c\rangle$, and write

$$\pi : X \longrightarrow Z, \qquad f : Z \longrightarrow B, \qquad P = \ker(\mathrm{Nm}_f)^0.$$

The quadratic extension is

$$\mathbf{Q}(Z) = \mathbf{Q}(B)(\sqrt{j - 1728}). \tag{6.1}$$

Indeed, order the three nonzero points of $E[2]$. The sign of their permutation is the squareclass of the discriminant of the cubic defining $E[2]$; after removal of a square factor this is $j - 1728$. Every cusp width at odd level is odd. Thus the sign cover is ramified at each cusp, with one point above it and the same residue field. The remaining ramification consists of the elliptic points of order two. A point above $j = 1728$ with trivial elliptic inertia has even valuation in (6.1).

A transposition b gives a quotient

$$\beta : X \longrightarrow Y = X/\langle b\rangle \simeq X_0(2M).$$

Choose it so that $\beta(\tau) = 2\tau$ in the usual coordinates. Both f and β have a rational ramified cusp. Their Jacobian pullbacks are injective: a nontrivial kernel class would be two-torsion and define an étale quadratic cover factoring through the given double cover. Equality of degrees would identify these covers, contradicting ramification. Duality also shows that $\ker \mathrm{Nm}_f$ is connected, so its rational points are those of P.

There are no Gaussian branch points if some prime dividing M is $3 \pmod 4$. Otherwise their closed orbits have residue field $\mathbf{Q}(i)$ and the Gaussian indices of Theorem 4.1. The Gaussian branch points of $Y \to B$ map with residue degree one: multiplication by i on $E[2]$ has the unique invariant line $\ker(1+i)$, so the additional order-two subgroup is defined over the same field as the Gaussian point. This includes every prime power in M.

6.2. **A cubic model for the Prym.**

Proposition 6.2. *For every finite unramified extension $K/\mathbf{Q}_2$,*

$$4P(K)[2^\infty] = 0.$$

Proof. Let R be the ring of integers of $K(\zeta_3)$, and let $k = R/2R$. Add full three-level structure with specified Weil pairing over R. If $3 \mid M$, select the component on which the order-three subgroup of the cyclic marking is the specified line. The odd cyclic marking is finite étale on the elliptic locus, since $E[M]$ is finite étale. This holds with unrestricted prime powers in M.

The full-two cover of the full-three curve is the modular model [26, §7.1]

$$X(6) : y^2 = x^3 + 1, \qquad h = \frac{x^3+4}{3x^2} : X(6) \longrightarrow X(3).$$

The point $Q = (0, 1)$ has order three. Translation by Q is

$$x' = \frac{2(1-y)}{x^2}, \qquad y' = -y + \frac{y-1}{x}(x - x'). \tag{6.2}$$

Its invariant functions are

$$x_V = \frac{x^3+4}{x^2}, \qquad y_V = \frac{y(x^3-8)}{x^3}, \qquad y_V^2 = x_V^3 - 27.$$

The last identity follows by expanding

$$(x^3+1)(x^3-8)^2 = (x^3+4)^3 - 27x^6 = x^9 - 15x^6 + 48x^3 + 64.$$

Thus this translation generates the modular A_3 action.

At the supersingular point the regular chart is

$$x = u(u+t), \qquad y = 1 + u^2(u+t), \qquad 2 = tu(u+t). \tag{6.3}$$

Substitution gives $y^2 - x^3 - 1 = u^2(u+t)\{2 - tu(u+t)\}$. The translation (6.2) becomes exactly

$$(u, t) \longmapsto (t, -u - t). \tag{6.4}$$

For example $x' = -tu$ and $(y'-1)/x' = 2/(tu) - u = t$. The third iterate is the identity and $tu(u+t)$ is preserved. The lower coordinate is $h = (u^2 + ut + t^2)/3$, since $t^2 = 4/x^2$ and $x = u^2 + ut$.

The special fiber consists of three ordinary components meeting at supersingular triples. After the odd finite étale pullback, each component is the Frobenius twist of the fine odd-level curve. Its function field is obtained by adjoining $\sqrt{h}$ to the lower function field. Since that field is separable over the h-line, h is not a square; the purely inseparable quadratic extension is the function field of one connected component. There is one triple for each supersingular point of the fine base.

The cusp charts are smooth discs $R[[v]]$ with lower parameter v^2 up to a unit. Full-two cusp inertia has order two; at level six these are $q^{1/6}$ and $q^{1/3}$. Adding odd cyclic level introduces only odd roots of the cusp parameter, whose coprime normalization is again a smooth disc. The three cusp lifts are permuted by A_3. Thus the listed triples are all the nonsmooth points of the special fiber.

Make the tame extension $\varpi^3 = 2$ and blow up (u, t, ϖ) at every triple. The ϖ-chart is $u = \varpi X$, $t = \varpi Z$, $1 = XZ(X+Z)$. Its exceptional compactification is

$$E_s : XZ(X+Z) = W^3. \tag{6.5}$$

In characteristic two its partial derivatives are Z^2, X^2, W^2, so it is a smooth cubic. Its attachment points are $[1:0:0]$, $[0:1:0]$, $[1:1:0]$. In the u-chart,

$$t = uz, \qquad \varpi = uw, \qquad w^3 = z(1+z).$$

At $z = 0, 1$ the derivative of the last equation is a unit, so it eliminates z and leaves the precise semistable node $\varpi = uw$. The t-chart treats the third attachment. These three charts cover the exceptional divisor.

The action (6.4) on this cubic is $\rho[X : Z : W] = [Z : X + Z : W]$. It is free: for $W \neq 0$ a fixed point would force $X = Z = X + Z = 0$, and for $W = 0$ it permutes the three attachment points. It cycles the three central components. On the generic fiber full-three level removes elliptic stabilizers and cusp inertia is two, so the A_3 action there is also free. The quotient of the whole semistable model therefore has one central component C and one elliptic tail $E'_s = E_s/\langle\rho\rangle$ at each supersingular point, attached at one node. The graph is a tree. Its Jacobian has good reduction over $R[\varpi]$. Contraction of an unstable genus-zero central component does not change this statement.

Inertia $\sigma(\varpi) = \zeta_3\varpi$ acts on the cubic by $[X : Z : W] \mapsto [X : Z : \zeta_3 W]$. It commutes with ρ and fixes the attachment points. Its induced automorphism on E'_s is nontrivial of order three and fixes the attachment origin. Otherwise it would be a deck translation ρ^a, and a fixed attachment would force $a = 0$. Thus $1 + \sigma + \sigma^2 = 0$ on $\mathrm{Jac}(E'_s)$.

On C, inertia is trivial. The map $C \to B_k$ is purely inseparable of degree two, so its norm is an isomorphism on rational ℓ-adic Tate modules for odd ℓ. Compact-type reduction identifies the rational Tate space of the fine Prym with the sum of the tail spaces. The fine Prym itself has good reduction over $R[\varpi]$, by the Néron–Ogg–Shafarevich criterion. Faithfulness of its rational Tate module gives $1 + \sigma + \sigma^2 = 0$ on its own good special fiber.

Edixhoven's filtration of the Prym's own Néron special fiber [19, Theorems 4.2 and 5.3] has a connected unipotent subgroup F^1 and quotient the σ-fixed subgroup of the good special fiber. The quotient is killed by three, because

$$(\sigma - 1)(\sigma^2 - 1) = 3 \quad \text{under } 1 + \sigma + \sigma^2 = 0.$$

The positive filtration is killed by four [19, Remark 5.4(2)]. In its jet description the coordinates lie in $(\varpi) \subset k[\varpi]/(\varpi^3)$. In characteristic two multiplication by two sends this ideal into its square, and multiplication by four into $(\varpi^4) = 0$. Thus every two-primary special point is killed by four.

The formal kernel over the original unramified ring has no two-primary torsion. To prove this, write $[2](T) = 2T + Q(T) + O(T^3)$. The reduced quadratic term is Frobenius-linear. The special identity group is unipotent and killed by some power of two, so comparison of the lowest-degree terms in the iterates of [2] makes $\bar{Q}$ nilpotent under composition. A nonzero order-two point of minimum coordinate valuation $v \geq 2$ is impossible since the linear term has valuation $v + 1 < 2v$. For $v = 1$, write $T = 2w$; dividing the equation by four and reducing gives $\bar{w} + \bar{Q}(\bar{w}) = 0$ with $\bar{w} \neq 0$, contradicting nilpotence. Thus specialization is injective on two-primary torsion, and the fine Prym has exponent at most four.

The fine-to-coarse cover of Z has effective group A_4 if $3 \nmid M$ and C_3 on the chosen component if $3 \mid M$. Reduction modulo three gives $\mathrm{SL}_2(\mathbf{F}_3)$ or its order-six upper Borel, respectively, followed by division by $\{\pm 1\}$. Both effective groups have odd abelianization. The kernel of Picard pullback injects into their character groups: a trivialization of a pulled-back bundle and its translates differ by constants on a proper connected curve. Hence there is no geometric two-primary kernel. Norm commutes with this pullback, as one checks on divisors away from the branch loci; the connected image of P lies in the fine Prym. Since $\mathbf{Q}_2(\zeta_3)$ is unramified, the fine exponent bound proves the assertion over K. □

6.3. **The first Gaussian correction.**

Lemma 6.3. *Every point of $P(\mathbf{Q})[2]$ is represented by a rational degree-zero cusp divisor on Z.*

Proof. If $M = 1$, the sign cover $Z \to X_0(1)$ is branched exactly at $j = 1728$ and the cusp ∞. Riemann–Hurwitz gives genus zero; its rational cusp identifies Z with $\mathbf{P}^1$ over $\mathbf{Q}$. Thus $P = 0$ and the assertion is immediate. Assume $M > 1$.

The involution ι of f acts by -1 on P, so $L \in P(\mathbf{Q})[2]$ is invariant. A rational cusp makes it a line bundle over $\mathbf{Q}$. Normalize $\iota^* L \simeq L$ to be the identity at a rational ramified cusp x_0. Its square is a constant equal to one there, and hence gives a linearization over $\mathbf{Q}$.

Let D be the reduced rational ramification divisor of negative stabilizer signs. Twisting by $\mathcal{O}(-D)$ makes inertia trivial. Quotient descent gives $L = f^*E \otimes \mathcal{O}(D)$ with E a line bundle on B. Since $2\deg E + \deg D = 0$, $\deg D$ is even. Put $y_0 = f(x_0)$ and $R = [E + (\deg D/2)y_0] \in J_B(\mathbf{Q})$. Then

$$L = f^*R + [D - \deg D\, x_0], \qquad 2R = -[f_*D - \deg D\, y_0]. \tag{6.6}$$

The second equality follows from $2L = 0$, $f^*f_*D = 2D$, $f^*y_0 = 2x_0$, and injectivity of f^*.

If a prime factor of M is $3 \pmod 4$, there are no Gaussian branch points, and D is already cuspidal. Otherwise all prime factors of M are $1 \pmod 4$. Modulo twice $J_B(\mathbf{Q})$ and rational cusps, the second equation is then the Gaussian obstruction of Theorem 4.1. Its kernel consists of constant patterns. If the Gaussian part of D is all one, reverse the entire linearization; its new negative divisor is the complementary ramification divisor, with no Gaussian part. If the pattern is zero, retain the linearization. In every case D is now cuspidal. Consequently $2R$ is torsion, so R is torsion. Taking two-primary parts and applying Theorem 3.1 makes R rationally cuspidal. Equation (6.6) proves the result. □

6.4. **Rational eta units and norms.** Write $\eta_\delta(\tau) = \eta(\delta\tau)$. A $\mathbf{Q}$-rational function at level L whose divisor is supported on cusps has the expression

$$u = \lambda \prod_{\delta \mid L} \eta_\delta^{e_\delta}, \qquad \lambda \in \mathbf{Q}^*, \quad e_\delta \in \mathbf{Z}. \tag{6.7}$$

For an eta quotient of weight zero with trivial character, the integral modularity conditions are

$$\begin{gathered} \sum_\delta e_\delta = 0, \qquad \sum_\delta \delta e_\delta \equiv 0 \pmod{24}, \qquad \sum_\delta (L/\delta) e_\delta \equiv 0 \pmod{24}, \\ \sum_\delta v_p(\delta) e_\delta \equiv 0 \pmod 2 \quad \text{for every } p \mid L. \end{gathered} \tag{6.8}$$

The eta transformation formula explains these conditions: the two sums modulo 24 cancel the translation multipliers at infinity and zero, and the valuation conditions make the quadratic character of $\prod \delta^{e_\delta}$ trivial. Completeness of (6.7) for rational units follows from [46, Theorem 7 and Corollaries 8–9]. Namely, multiply by a power of Δ to remove all cusp poles, and by a rational scalar to clear the bounded Fourier denominators. The resulting holomorphic congruence form has integral coefficients and no zero in the upper half-plane, so that theorem gives an integral eta quotient. Remove the power of Δ.

At $1/c$ on $X_0(L)$, $c \mid L$, put $m = L/c$, $h = \gcd(c, m)$, $w = m/h$. The cusp residue field is $\mathbf{Q}(\zeta_h)$. Choose $a, b \in \mathbf{Z}$ with $a(c/h) + bw = 1$. In the chart $\gamma = \left(\begin{smallmatrix} 1 & 0 \\ c & 1 \end{smallmatrix}\right)$, put $q_c = \exp(2\pi i \tau/w)$. A parameter defined over the cusp residue field is

$$t = \zeta_m^a q_c. \tag{6.9}$$

To verify the field of definition, write the Tate cyclic marking as $z = \zeta_L q_E^{c/L}$. Then $t = (z^c)^a q_E^b$ and $q_E = \zeta_h^{-a} t^w$, $z^c = \zeta_h^b t^{c/h}$. The latter two expressions recover the marked Tate curve over $\mathbf{Q}(\zeta_h)$. Different Bezout choices change t by a unit of that field.

For η_δ, put $k = \gcd(c, \delta)$, $C = c/k$, $D = \delta/k$, and choose d with $Dd \equiv 1 \pmod C$. Its analytic phase and positive amplitude are

$$\phi(c, \delta) = \frac{D + 1/D}{24C} - \frac{s(d, C)}{2} - \frac{1}{8}, \qquad \left(\frac{k}{\delta}\right)^{1/2}. \tag{6.10}$$

Its q_c-order is $wk^2/(24\delta)$. Here the Dedekind sum is

$$s(d,C)=\sum_{j=1}^{C-1}\left(\left(\frac{j}{C}\right)\right)\left(\left(\frac{dj}{C}\right)\right),\qquad ((x))=\begin{cases}x-\lfloor x\rfloor-\frac12, & x\notin\mathbf{Z},\\ 0, & x\in\mathbf{Z}.\end{cases}$$

In particular $s(d,1)=0$. For an explicit derivation, choose $\left(\begin{smallmatrix}D & b_0\\ C & d\end{smallmatrix}\right)\in \mathrm{SL}_2(\mathbf{Z})$. The transformed argument is $(k/D)\tau-b_0/D$. The eta multiplier contributes $(D+d)/(24C)-s(d,C)/2-1/8$, and the translation contributes $-b_0/(24D)$. Since $Dd-Cb_0=1$, their sum is (6.10). The common square root of $c\tau+1$ cancels for a weight-zero product. Consequently the leading coefficient of (6.7) in the parameter (6.9) is

$$\lambda\exp\left(2\pi i\left[\sum_{\delta}e_\delta\phi(c,\delta)-\mathrm{ord}_{1/c}(u)\frac{a}{m}\right]\right)\prod_{p\mid L}p^{\frac12\sum_\delta e_\delta(v_p(\gcd(c,\delta))-v_p(\delta))}. \tag{6.11}$$

Let g,u be nonzero rational functions on a smooth curve over a number field. At a closed point x with residue field $k(x)$, choose a local parameter z and write

$$g=g_xz^m(1+O(z)),\qquad u=u_xz^n(1+O(z)),\qquad g_x,u_x\in k(x)^*,\quad m=\mathrm{ord}_x(g),\quad n=\mathrm{ord}_x(u).$$

If u is a function norm from the quadratic extension defined by $\sqrt{g}$, then

$$(-1)^{mn}g_x^nu_x^{-m}\in k(x)^{*2}. \tag{6.12}$$

Identify the completed local field with $k(x)((z))$. Its characteristic is zero, so a unit with constant term one has a square root with constant term one. Removing this unit square and an even power of z from g preserves the quadratic extension. If m is odd, it has an equation $t^2=g_xz$ and $\mathrm{Nm}(t)=-g_xz$. A norm witness has the form $t^n(a+O(t))$, with $a\in k(x)^*$, and therefore

$$u_x=(-g_x)^na^2.$$

Substitution into (6.12) gives $g_x^{n(1-m)}a^{-2m}$, a square because $1-m$ is even. Negative values of m and n are allowed in these Laurent expansions. If m is even and g_x is a square, every factor in (6.12) is a square. If m is even and g_x is not a square, the completed extension is the unramified quadratic extension

$$k(x)(\sqrt{g_x})((z))/k(x)((z)).$$

Its norms have even valuation, so n is even, and the displayed expression is again a square. Finally, replacing z by a parameter with leading coefficient $\alpha\in k(x)^*$ changes g_x,u_x to $g_x\alpha^{-m},u_x\alpha^{-n}$. These factors cancel in (6.12), as do the parameter corrections in (6.11).

Lemma 6.4. *Let $\mathcal{U}_L$ denote the group of rational cusp units at level L, including $\mathbf{Q}^*$. For odd M,*

$$\mathcal{U}_{2M}\cap\mathrm{Nm}_\beta\,\mathbf{Q}(X_0(4M))^*=\mathrm{Nm}_\beta\,\mathcal{U}_{4M}.$$

Proof. Put $H=\eta_1^{24}/\eta_2^{24}$ on $X_0(2)$ and $w=\eta_1^8/\eta_4^8$ on $X_0(4)$. The exact identities are

$$\beta^*H=w(w+16),\qquad \beta^*(H+64)=(w+8)^2. \tag{6.13}$$

The eta identity behind them is

$$\eta_2^{24}=\eta_1^{16}\eta_4^8+16\eta_1^8\eta_4^{16}.$$

For $\mathrm{Im}\,z>0$, define the Jacobi theta constants by

$$\vartheta_2(z)=\sum_{n\in\mathbf{Z}}e^{\pi i(n+1/2)^2z},\quad \vartheta_3(z)=\sum_{n\in\mathbf{Z}}e^{\pi in^2z},\quad \vartheta_4(z)=\sum_{n\in\mathbf{Z}}(-1)^ne^{\pi in^2z}.$$

Substitute $\vartheta_4(2\tau)=\eta_1^2/\eta_2$, $\vartheta_2(2\tau)=2\eta_4^2/\eta_2$, $\vartheta_3(2\tau)=\eta_2^5/(\eta_1^2\eta_4^2)$ in Jacobi's $\vartheta_3^4=\vartheta_2^4+\vartheta_4^4$ and multiply by $\eta_1^8\eta_2^4\eta_4^8$. Since $g=H+64$ has order -1 at infinity, the quadratic function field is obtained by adjoining $\sqrt{g}$.

With upper parameter t and lower parameter $q = t^2$, the two roots of $z(z+16) = H(t^2)$ are distinguished by their leading terms. Thus $w(-t) = -16 - w(t)$, and

$$t(w(t)+8) = V(t^2), \qquad V \in \mathbf{Z}_2[[q]]^*, \quad V(0) = 1.$$

Hence $g = q^{-1}V(q)^2$. Let $R = \widehat{\mathbf{Z}_2[[q]]_{(2)}}$ and $F = \operatorname{Frac} R$. The vertical extension is $F(t)$, $t^2 = q$. Its integral ring is $R[t]$: its reduction is the field $\mathbf{F}_2((q))(\sqrt{q})$ and its maximal ideal is generated by two, so it is a DVR. Its trace is contained in $2R$. If a norm W belongs to $1 + 2R$, its witness has residue one because the residue norm is the square map. Writing it as $1 + 2z$ gives

$$\operatorname{Nm}(1+2z) = 1 + 2\operatorname{Tr}(z) + 4\operatorname{Nm}(z) \equiv 1 \pmod{4R}. \tag{6.14}$$

Write $u = \lambda \prod_{d|M} \eta_d^{b_{0,d}} \eta_{2d}^{b_{1,d}}$ and $A = \operatorname{ord}_\infty u$. The ramified rational infinity cusp forces $\lambda = (-1)^A a^2$ for some $a \in \mathbf{Q}^*$. Remove the scalar norm a^2. Put $E_d(t) = \prod_{\nu \geq 1}(1 - t^{d\nu})$ and $B_d = b_{0,d} + 2b_{1,d}$. For odd d the exact Euler norm is

$$E_d(t)E_d(-t) = E_d(q)^3 / E_{2d}(q). \tag{6.15}$$

Separate the even indices, contributing $E_d(q)^2$, and the odd indices, contributing $E_d(q)/E_{2d}(q)$. First take this identity in the finite free power-series algebra $\mathbf{Z}_2[[t]]/\mathbf{Z}_2[[q]]$; the determinant norm is continuous in the parameter topology. Then base change to $R[t]/R$.

Dividing u by the norm of $t^A \prod_d E_d(t)^{B_d}$ gives

$$W = \prod_d (E_d(q)^2 / E_{2d}(q))^{-b_{0,d} - 3b_{1,d}}.$$

It is a norm in $1 + 2R$, hence is one modulo four. For any positive integer m,

$$\frac{E_m(q)^2/E_{2m}(q) - 1}{2} \equiv \sum_{\nu \geq 1} q^{m\nu^2} \pmod 2. \tag{6.16}$$

Indeed the product is $\prod_\nu (1 - q^{m\nu})/(1 + q^{m\nu})$. Its first-order expansion modulo four counts divisors of each integer, an odd number exactly for squares. A finite collection of these series with different m is independent, since its least active m gives the unique lowest term. Therefore

$$b_{0,d} \equiv b_{1,d} \pmod 2 \qquad (d \mid M). \tag{6.17}$$

The identity $4R \cap \mathbf{Z}_2[[q]] = 4\mathbf{Z}_2[[q]]$ justifies this coefficient test after vertical localization and completion.

At the lower rational cusp $1/2$, the width is M and g has order $-M$ and leading coefficient one. In fact $\phi(2,1) = -1/12$ and $\phi(2,2) = -1/24$, so H has phase $24(-1/12 + 1/24) = -1$ and amplitude one; the parameter correction for order $-M$ is also one. The constant 64 does not alter its leading term. For each odd $p \mid M$, formula (6.11) gives the valuation of the leading coefficient of u as $-S_p/2$, where

$$S_p = \sum_d v_p(d)(b_{0,d} + b_{1,d}).$$

Lower modularity makes S_p even. The square condition (6.12) then gives $S_p \equiv 0 \pmod 4$, since M is odd.

The exact exponent norm on each odd axis is $b_0 = 3a_0 + 2a_1$, $b_1 = -a_0 + 2a_2$. All its integer solutions are

$$a_{0,d} = -b_{1,d} + 2z_d, \quad a_{1,d} = (b_{0,d} + 3b_{1,d})/2 - 3z_d, \quad a_{2,d} = z_d, \qquad z_d \in \mathbf{Z}. \tag{6.18}$$

Their weight is half the target weight, and $a_0 + 2a_1 + 4a_2 = b_0 + 2b_1$. Thus weight and the infinity Newman congruence hold. Every odd character moment is $S_p/2$, hence even. The dyadic character moment is even if $\sum z_d$ is even, since the lower condition makes $\sum b_{1,d}$ even.

For the reciprocal congruence define

$$P_0 = \sum_d d(b_{0,d} + 2b_{1,d}), \quad F_0 = \sum_d (M/d)(2b_{0,d} + b_{1,d}),$$
$$W_0 = \sum_d (M/d)b_{1,d}, \qquad T_0 = \sum_d (M/d)(b_{0,d} - b_{1,d}).$$

Lower modularity gives $24 \mid P_0, F_0$. Using $d^{-1} \equiv d \pmod 4$ and $M^2 \equiv 1 \pmod 4$,

$$2P_0 - MF_0 \equiv 3\sum_d db_{1,d} \pmod 4,$$

so $4 \mid W_0$. The exact relation $2T_0 = F_0 - 3W_0$ implies $6 \mid T_0$. The source reciprocal sum is $T_0 + 3\sum_d (M/d)z_d$. Choose $z_d = 0$ for $d \neq M$ and the even value $z_M \equiv -T_0/3 \pmod 8$. Every congruence in (6.8) now holds. The upper eta product is a rational unit. Its norm has the target exponents and the sign $(-1)^A$. Multiplication by the rational scalar a restores λ, proving the exact norm equality. □

6.5. **The second Gaussian correction.**

Lemma 6.5. *If $a \in J_X(\mathbf{Q})[2^\infty]$ satisfies $(1-b)a \in C_{\mathrm{rat}}(X)$ and $\mathrm{Nm}_\beta\, a \in C_{\mathrm{rat}}(Y)$, then*

$$a \in \beta^* J_Y(\mathbf{Q})[2^\infty] + C_{\mathrm{rat}}(X)[2^\infty].$$

Proof. For $M = 1$, the curve $X = X_0(4)$ has genus zero, so $a = 0$ and the assertion is immediate. Assume $M > 1$.

Represent a by a rational degree-zero divisor A. Let D_0 be a rational cusp divisor representing $(1-b)a$. Choose h_0 with $\mathrm{Div}(h_0) = D_0 - (1-b)A$. Its norm is a lower cusp unit. Lemma 6.4 gives an upper cusp unit v with $\mathrm{Nm}\, v = \mathrm{Nm}\, h_0$, including the constant. Hilbert 90 gives $h_0/v = t/(bt)$. After replacing A by $A + \mathrm{Div}(t)$, its part away from cusps is b-invariant.

Unramified invariant divisors descend integrally, including at inert closed points, where the ramification index is one. At a ramified point split any signed coefficient as $n = 2\lfloor n/2 \rfloor + \epsilon$, $\epsilon = 0$ or 1. Thus

$$A = \beta^* F + D_{\mathrm{G}} + C, \tag{6.19}$$

where F is rational on Y, C is a rational cusp divisor, and D_{G} is a reduced rational divisor of Gaussian branch points. Using rational ramified cusps x_0, y_0, set $R = [F - \deg F\, y_0]$ and $r_{\mathrm{G}} = [D_{\mathrm{G}} - \deg D_{\mathrm{G}}\, x_0]$. The degree correction leaves a rational degree-zero cusp class, so $a = \beta^* R + r_{\mathrm{G}} + c$.

If $D_{\mathrm{G}} = 0$, (6.19) already has the required form. This includes every M with a prime factor $3 \pmod 4$. If $D_{\mathrm{G}} \neq 0$, all prime factors of M are $1 \pmod 4$. Taking norms and pushing to B puts its Gaussian pattern in the kernel of Theorem 4.1. The residue degrees are one, as checked above. The pattern is consequently constant and, being nonzero, is all one. In this case use the function $h = w + 8$ in (6.13). Its order is odd at every interior ramification point; at all other interior points its divisor is a pullback. Hence

$$\mathrm{Div}(h) = D_{\mathrm{G}} + \beta^* G + C_0$$

for a rational divisor G on Y and a cusp divisor C_0. Subtracting this principal divisor gives $A \sim \beta^*(F - G) + C - C_0$.

In either case, for a lower divisor of degree d, replace it by that divisor minus dy_0, and the cusp remainder by that remainder plus $2dx_0$. Since $\beta^* y_0 = 2x_0$, the identity is preserved and both classes have degree zero. We obtain $a = \beta^* R' + c'$. Manin–Drinfeld makes c' torsion, and injectivity of β^* makes R' torsion. Taking two-primary parts proves the assertion. □

6.6. **The three-old quotient.**

Proof of Theorem 6.1. Division by three is an automorphism of a finite two-primary group. Put $e = (1 + c + c^2)/3$. For $v \in (1 - e)T_X$ and the three transpositions s_0, s_1, s_2,

$$\sum_{i=0}^{2}(1 + s_i)v = 3v + (1 + c + c^2)s_0v = 3v.$$

Each $(1 + s_i)v$ is the pullback of its norm to $X/\langle s_i\rangle$. This summand lies in the sum of the three old images. The maps π^* and $\mathrm{Nm}_\pi/3$ identify eT_X with T_Z.

For $z \in T_Z$, its norm to B is cuspidal by Theorem 3.1. Every cusp of B has one lift with the same residue field, so norm is surjective on rational degree-zero cusp divisors. Subtract a cusp class with that norm and take its two-primary part. The remainder is $p \in P(\mathbf{Q})[2^\infty]$. By Proposition 6.2 and Lemma 6.3, $2p$ is cuspidal.

For $a = \pi^*p$, the transposition b acts by -1. Thus $(1 - b)a = 2a$ is cuspidal, and $\beta^* \mathrm{Nm}_\beta a = (1 + b)a = 0$. Injectivity of pullback gives $\mathrm{Nm}_\beta a = 0$ exactly. Lemma 6.5 puts a in an old image plus rational cusps. This proves the three-old surjection. Theorem 5.1 kills its source. □

7. The dyadic induction

Theorem 7.1. *Let M be odd and $n \geq 3$. If $T_{2^{n-1}M} = C_{2^{n-1}M}$, then $T_{2^nM} = C_{2^nM}$. Consequently $T_{2^nM} = C_{2^nM}$ for every $n \geq 0$.*

Put $N = 2^nM$, $L = N/2$, $X = X_0(N)$, and $Y = X_0(L)$. The two maps $\alpha(\tau) = \tau$ and $\beta(\tau) = 2\tau$ have deck involutions

$$A = \begin{pmatrix} 1 & 0 \\ N/2 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix}.$$

They are defined over $\mathbf{Q}$ and commute modulo $\Gamma_0(N)$. Write $D = (1 - A)(1 - B)$ for their group-ring action on Picard groups. Since $4 \mid L$, the lower curve has no order-two elliptic stabilizer; a quadratic cover cannot branch at an odd-order stabilizer. Both covers are therefore unramified off cusps in characteristic zero. The cusp zero is rational and totally ramified for α, and infinity is rational and totally ramified for β. In the latter case the exact parameter relation is $q = t^2$.

7.1. **The regular fine model.** Work over an unramified dyadic DVR R containing ζ_3, with residue field $k = R/2R$. Add a full-three basis with fixed Weil pairing, and, if $3 \mid M$, choose the component with the specified order-three line. Let X' be the resulting connected fine curve and $h : X' \to X$ the forgetful map. Its effective group is A_4 when $3 \nmid M$ and C_3 on the selected component when $3 \mid M$. The character argument in Proposition 6.2 shows

$$h^* : J_X[2^\infty] \hookrightarrow J_{X'}[2^\infty]. \tag{7.1}$$

Fix the normalization $\mathscr{X}'$ of $\mathbf{P}^1_R$ in this fine function field. It is finite over the projective j-line. On the elliptic locus it is the fine cyclic-subgroup moduli scheme. Katz–Mazur's regularity theorem for cyclic subgroups [25, Theorem 6.6.1], followed by the finite étale full-three rigidification, makes this locus regular of dimension two, including its supersingular points. The compactification is the normalization of [25, §§8.6.2–8.6.3].

We also verify regularity at cusps on this same model. Write Q for a Tate parameter tending to zero. After extending coefficients to the Witt ring $W = W(\overline{\mathbf{F}}_2)$, the normalized mixed cusp rings of connected height $a \geq 1$ and étale height $b \geq 1$, with $a + b = n$, have the forms

$$\begin{cases} W[\zeta_{2^b}][[Q]], & a \geq b, \\ W[\zeta_{2^a}][[t]], \quad Q = \zeta t^{2^{b-a}}, & a < b. \end{cases}$$

Here ζ is a root of unity in the coefficient ring. The pure cases are $W[[Q]]$ and $W[[Q^{1/2^n}]]$. Their coefficient rings are DVRs, hence these are regular two-dimensional rings. This is precisely the all-prime Tate normalization in [18, §1.2, Proposition 1.2.2.1 and Theorem 1.2.3.1]. Odd level adds odd roots of unity and roots of a parameter of odd order. Normalizing with coprime exponents again gives a power-series ring over the same dyadic cyclotomic DVR. Full-three level adds $\zeta_3, Q^{1/3}$. If an existing cusp chart is $O_c[[t]]$ with $Q = \zeta t^w$, its normalized fine chart is $O_c[[u]]$ with $t = u^{3/\gcd(3,w)}$ up to a unit, after an unramified extension for the odd roots of unity. Cubing is invertible on the dyadic roots of unity. Selection of a marking component when $3 \mid M$ selects some of these rings. At the raw marked Tate-curve level simultaneous negation of the basis pairs distinct branches; its parameter monodromy has order dividing three and cannot identify a basis with its negative. Quotienting these pairs leaves the same regular rings. On the fine moduli curve the central $-I$ already acts trivially.

The lifts of the deck transformations are

$$A' = \begin{pmatrix} 1 & 0 \\ 3N/2 & 1 \end{pmatrix}, \qquad B' = \begin{pmatrix} 1 & 3/2 \\ 0 & 1 \end{pmatrix}. \tag{7.2}$$

Their squares lie in $\Gamma_0(N) \cap \Gamma(3)$, and direct multiplication gives

$$[A', B'] = \begin{pmatrix} 1 - 9N/4 & 27N/8 \\ -27N^2/8 & 1 + 9N/4 + 81N^2/16 \end{pmatrix}. \tag{7.3}$$

For $N = 8m$ this is $\left(\begin{smallmatrix} 1-18m & 27m \\ -216m^2 & 1+18m+324m^2 \end{smallmatrix}\right)$, in the fine subgroup. The transformations normalize that subgroup. For example conjugation by B' replaces the upper-right entry of $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ by $b + \frac{3}{2}(d-a) - \frac{9}{4}c$; a, d are odd and $8 \mid c$, so it is integral. The other entries and the conjugation by A' give the same divisibility requirements, and reduction modulo three is unchanged. Their squares are $\left(\begin{smallmatrix} 1 & 0 \\ 3N & 1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 3 \\ 0 & 1 \end{smallmatrix}\right)$. Thus they are commuting involutions on the fine curve.

The full-three marking is held fixed exactly by these deck maps. With $P = \tau/3, Q = 1/3$ on $\mathbf{C}/(\mathbf{Z}\tau + \mathbf{Z})$, the isogeny $\phi : z \mapsto 2z$ defining β sends the marking to $(\phi P, 2\phi Q)$. The second factor restores the specified pairing because $2 \cdot 2 = 1$ modulo three. The lower marking is $(2\tau/3, 1/3)$. Since $\beta B'(\tau) = 2\tau + 3$, it is unchanged. The integral $A' \equiv I \pmod 3$ fixes the source marking for α. A degree-two isogeny is an isomorphism on odd torsion, so it transports all cyclic M-markings uniquely.

Both involutions extend to $\mathscr{X}'$ itself. For A' this follows from $j(A'\tau) = j(\tau)$ and normalization. For B' we give a finite-graph argument across all cusps. The integral degree-two modular polynomial is

$$\begin{aligned} \Phi_2(X, Y) = {} & X^3 + Y^3 - X^2Y^2 + 1488XY(X + Y) \\ & - 162000(X^2 + Y^2) + 40773375XY \\ & + 8748000000(X + Y) - 157464000000000. \end{aligned}$$

Its bihomogenization restricts at the two infinity divisors to $X_1^3Y_0^3$ and $X_0^3Y_1^3$. It is monic in each affine variable; the displayed boundary equations show finite fibers there as well. Thus its projective correspondence has finite projections. The composite of two copies also has finite projections. Now $(j(\tau), j(2\tau))$ and $(j(2\tau), j(B'\tau))$ are on this correspondence. The closure of the graph of B' in $\mathscr{X}' \times_R \mathscr{X}'$ is consequently finite over either factor, since $\mathscr{X}'$ is finite over the j-line. A finite birational map to the normal $\mathscr{X}'$ is an isomorphism. This extends B' to the whole model. The involution and commutativity relations persist by separatedness.

Every vertical component of this finite-j model dominates $\mathbf{P}^1_k$: a finite map cannot send a one-dimensional component to a point. Thus every vertical generic point is ordinary. The entire closed fiber $Y' = (2 = 0) \subset \mathscr{X}'$ is a Cartier divisor in a regular surface. It is Cohen–Macaulay of pure dimension one, with no embedded associated point. Its local rings therefore inject into the product of their localizations at the generic points, retaining the complete Artinian thickness.

7.2. **Ordinary charts and the Picard identity.** We now base change the model to W and write $k = \overline{\mathbf{F}}_2$. For a cyclic subgroup of connected height a and étale height b, $a+b=n$, $a,b \geq 1$, the full ordinary completed ring is

$$R_{a,b} = W[[q-1, z-1]]/(z^{2^{b-1}} + q^{2^{a-1}}). \tag{7.4}$$

Here q, z are the source and target Serre–Tate units. In the torsion presentation the subgroup contains μ_{2^a} and is generated modulo it by $(x, 1/2^b)$. Set $z = x^{2^a}$. Exact cyclic order is the cyclotomic condition $\Phi_{2^a}(x^{2^b}/q) = 0$, which is (7.4) [25, Theorem 13.6.6]. Squaring this equation gives the isogeny relation $z^{2^b} = q^{2^a}$. The pure endpoint rings are

$$R_{n,0} = W[[q-1]],\ z = q^{2^n}, \qquad R_{0,n} = W[[z-1]],\ q = z^{2^n}.$$

For $a \geq b$, set $\xi = z/q^{2^{a-b}}$; then $\xi^{2^{b-1}} = -1$ and $R_{a,b} = W[\zeta_{2^b}][[q-1]]$. For $a < b$, set $\xi = q/z^{2^{b-a}}$ and obtain $W[\zeta_{2^a}][[z-1]]$. All changes are invertible since these parameters are units. If $d = \min(a,b) \geq 1$, the whole closed ring is

$$R_{a,b}/2 = k[[T]][\epsilon]/(\epsilon^{2^{d-1}}), \qquad \epsilon = \xi - 1. \tag{7.5}$$

For α, the standard order-2^{n-1} subgroup gives $(q,z) \mapsto (q, z^2)$. If $b \geq 2$, its nontrivial deck map is $A'(q) = q$, $A'(z) = -z$, also at the pure étale endpoint. At the lower pure connected endpoint the two upper branches are

type	upper ring	source and target
$(n,0)$	$W[[q-1]]$	$q,\ z = q^{2^n}$
$(n-1,1)$	$W[[q-1]]$	$q,\ z = -q^{2^{n-2}}$.

Both lower maps are the identity on q; A' exchanges them.

For β, when $a \geq 2$ the map is $(q,z) \mapsto (q^2, z)$ and $B'(q) = -q$, $B'(z) = z$. Its action in the cyclotomic coordinates is

range	$B'(T)$	$B'(\xi)$
$a > b$	$-T-2$	ξ
$a = b$	$-T-2$	$-\xi$
$a < b$	T	$-\xi$.

For A' with $b \geq 2$, it fixes q and negates ξ when $a \geq b$; when $a < b$ it sends $z-1$ to $-(z-1)-2$ and fixes ξ. In every case each generator is fixed modulo the entire ideal (2), including ϵ and all its powers in (7.5). Thus B' is the identity on every full closed chart of connected height at least two, and A' on every full closed chart of étale height at least two.

For the exceptional β pair at lower type $(0, n-1)$, use $W[[z-1]]$ with $q_{\mathrm{low}} = z^{2^{n-1}}$. Its upper branches are

type	upper ring	q_{up}
$(0,n)$	$W[[z-1]]$	z^{2^n}
$(1,n-1)$	$W[[z-1]]$	$-z^{2^{n-2}}$.

(7.6)

The first is the étale predecessor $q_{\mathrm{up}} = q_{\mathrm{low}}^2$; the second is the connected predecessor $q_{\mathrm{up}}^2 = q_{\mathrm{low}}$ satisfying cyclicity. The other sign in the connected equation is not cyclic. Both lower maps are the identity on z, and B' exchanges these branches. Both have multiplicity one. The exceptional connected heights for B' are therefore $\{0,1\}$, and those for A' are $\{n-1, n\}$. They are disjoint already for $n=3$; intermediate heights are fixed by both on their whole closed rings.

Form the finite quotient only after reduction:

$$\pi_0 : Y' \longrightarrow Q = Y'/\langle A', B' \rangle.$$

For a local unit f on $\pi_0^{-1}V$, the preceding full generic-stalk calculation gives

$$f\, A'B'f = A'f\, B'f. \tag{7.7}$$

At a generic stalk where A' is the identity, $A'f = f$, $A'B'f = B'f$; at a stalk where B' is the identity, commutativity gives $B'f = f$, $A'B'f = A'f$. Injection into the full generic localizations proves (7.7) everywhere, including supersingular points and cusps.

For a finite morphism π_0, $R^1(\pi_0)_*\mathbf{G}_m = 0$ on the Zariski site. At $q \in Q$, choose an affine neighborhood $\operatorname{Spec} A$ with inverse image $\operatorname{Spec} B$, B finite over A. The ring $B \otimes_A A_q$ is semilocal, so every invertible module is free. Its trivialization and inverse descend to a smaller neighborhood by finite presentation. Hence the stalk of the displayed R^1 is zero. Leray gives a natural equivariant isomorphism

$$\operatorname{Pic}(Y') = H^1(Q, (\pi_0)_*\mathbf{G}_m).$$

The multiplicative sheaf endomorphism in (7.7) is trivial; thus

$$(1 - A'^*)(1 - B'^*) \operatorname{Pic}(Y') = 0. \tag{7.8}$$

The same argument applies to the finite map obtained after any k-base change: tensoring the generic-localization injection over the field preserves it. Thus (7.8) also holds on the relative Picard functor, including all nilpotent contributions.

For $P \in T_N$, a line bundle representing h^*P extends to this same regular model: the horizontal closure of a generic divisor is Cartier. Applying $(1 - A'^*)(1 - B'^*)$ to it gives an extension of h^*DP trivial on the entire Y'. Regular-model Picard–Néron comparison, with the smooth multiplicity-one cusp section, puts this point in the formal kernel. Over an unramified dyadic ring such two-primary torsion is killed by two: doubling takes the $2R$ disc to $4R$, where the linear term of further doublings has strictly least valuation. By (7.1),

$$2DP = 0. \tag{7.9}$$

7.3. **The based quadratic cover.** Put $S = DP$. If it is nonzero, use its based quadratic cover, split at infinity. The fine line bundle $\mathscr{L}$ above has trivial whole closed fiber and square-trivial generic fiber. The divisor V of its generic square trivialization is vertical and has intersection zero with every component. The kernel of the vertical intersection matrix on a proper regular model is the line of the whole fiber F. Because a smooth cusp meets a multiplicity-one component, $V = aF$ with $a \in \mathbf{Z}$. The based square value at that cusp has even valuation, so a is even. Removing V by a square preserves the original based cover and gives $\mathscr{L}^2 \simeq \mathcal{O}$ on the same model.

Take frames with transitions $u_{ij} = 1 + 2a_{ij}$. Their squares are one modulo four. The local square-trivialization functions therefore glue modulo four to a global function. Cohomological flatness gives $H^0(\mathscr{X}'_{R/4}, \mathcal{O}) = R/4$; for this regular model with section it follows from [3, Proposition 3.3]. Its based value is one, so the functions are $s_i = 1 + 4c_i$. The algebras

$$\mathcal{O}[z_i]/(z_i^2 + z_i - c_i), \qquad z_j = a_{ij} + (1 + 2a_{ij})z_i \tag{7.10}$$

glue. The exact calculation is

$$(a + (1 + 2a)z)^2 + (a + (1 + 2a)z) = (1 + 2a)^2(z^2 + z) + a(a + 1).$$

On each algebra $(2z_i + 1)^2 = 1 + 4c_i$ is a unit, so the derivative is a unit. This is a finite étale cover of the entire fine model, with generic identification $w_i = 1 + 2z_i$ with the original based square-root cover.

At two, this integral based branch has parameter $q^{1/3}$ after the fine marking. Its intersection with the original rational Laurent field gives integrality in the original parameter q, since exponents and coefficients are unchanged by the tame refinement. At every odd prime the dual isogeny defining this same based cover has a companion whose composition is [2]; their integral formal tangent matrices are therefore invertible. These local branches satisfy Lemma 2.2. The original cover is congruence. It is unramified on the compact curve, so the compact-width argument of Theorem 2.3 gives

$$S \in \Sigma(N)[2]. \tag{7.11}$$

Write its even quadratic character field as $\mathbf{Q}(\sqrt{d})$ with $d > 0$ squarefree. The natural generator cover has fiber squareclasses $d, 1$ at infinity and zero, respectively, by the multiplicative and étale Tate cusp descriptions. The cover based at infinity therefore has classes $1, d$. Its extension (7.10) is étale also at the fine cusp zero, which is defined over $\mathbf{Q}(\zeta_3)$. Thus $\mathbf{Q}_2(\zeta_3)(\sqrt{d})$ is unramified over the unramified quadratic extension of $\mathbf{Q}_2$. The dyadic squareclass criterion gives

$$d \text{ odd}, \qquad d \equiv 1 \pmod 4. \tag{7.12}$$

Indeed an even squarefree radicand gives odd valuation, and an odd radicand 3 (mod 4) has ramified quadratic discriminant; neither ramification is removed by an unramified extension. Now the discriminant equals d, so the character conductor divides the odd part M. Since $4 \mid L$, there is no order-two elliptic stabilizer at level L; odd stabilizers and cusp parabolics are killed by this even quadratic character. It gives a compact lower Shimura class S_0. Conjugation by $\mathrm{diag}(2, 1)$ preserves the diagonal entries modulo odd conductors, hence

$$\beta^* S_0 = S.$$

We need one branch fact with its square trivialization retained. For a double cover $f : V \to W$ with a rational ramified basepoint, suppose line bundles satisfy $L_0^2 = f^* M_0$. Then $E = M_0/(\mathrm{Nm}_f L_0)$ has a specified square trivialization. Its based quadratic cover splits at every ramified point whose unique preimage has the same residue field. On its scheme fiber the algebra is a length-two local algebra $k[\epsilon]/(\epsilon^2)$, and multiplication by $a + b\epsilon$ has determinant a^2. Thus the norm fiber of L_0 is its fiber square, and the induced square trivialization of E has square value. The rational basepoint fixes its global twist.

Apply this to $Q = (1 - A)P$. Since $\beta^* \mathrm{Nm}_\beta Q = Q + BQ = 2Q - S$, we have

$$2Q = \beta^*(\mathrm{Nm}_\beta Q + S_0).$$

The branch fact therefore makes the based S_0-cover split at every same-field ramification point of β. Besides infinity, such a point is the rational cusp $1/2^n$, mapping to $1/2^{n-1}$. The exact matrix equality

$$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 2^n & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2^{n-1} & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

gives the parameter relation $q = t^2$. Forgetting the two-power part sends these two lower cusps to infinity and zero on $X_0(M)$. The fiber squareclasses are consequently $1, d$. Splitting forces d to be a rational square, so $S_0 = S = 0$. For $M = 1$, odd conductor alone already forces $d = 1$. Thus

$$(1 - A)(1 - B)T_N = 0 \qquad (n \geq 3). \tag{7.13}$$

7.4. **Norms at higher dyadic levels.**

Lemma 7.2. *For $r \geq 3$, M odd, and the scaling map $\beta : X_0(2^r M) \to X_0(2^{r-1} M)$,*

$$\mathcal{U}_{2^{r-1}M} \cap \mathrm{Nm}_\beta \mathbf{Q}(X_0(2^r M))^* = \mathrm{Nm}_\beta \mathcal{U}_{2^r M}.$$

The equality includes rational constants.

Proof. Put $s = r - 1$ and $\ell = 2^s M$. On the lower curve take

$$g = 16\eta_1^8 \eta_4^{16}/\eta_2^{24}; \qquad h = 4\eta_2^4 \eta_8^8/\eta_4^{12} \quad \text{on the upper curve.} \tag{7.14}$$

The exponents satisfy (6.8), and $h^2 = \beta^* g$ exactly. The infinity order of g is one, so adjoining $\sqrt{g}$ gives the quadratic function field. Moreover

$$g/16 = q(E_1^4 E_4^8/E_2^{12})^2.$$

Its vertical completion is therefore the same $t^2 = q$ extension as in Lemma 6.4, with integral trace in $2R$. The norm congruence (6.14) applies.

Use (6.7) to write

$$u = \lambda \prod_{\substack{d \mid M \\ 0 \le i \le s}} \eta_{2^i d}^{b_{i,d}}, \qquad A = \frac{1}{24} \sum_{i,d} 2^i d\, b_{i,d}.$$

If it is a function norm, the rational ramified infinity cusp gives $\lambda = (-1)^A a^2$. Remove a^2. Put $B_d = \sum_i 2^i b_{i,d}$ and divide by the norm of the local Laurent unit $t^A \prod_d E_d(t)^{B_d}$. By (6.15), the quotient W has exponent vectors

$$b'_{0,d} = b_{0,d} - 3B_d, \quad b'_{1,d} = b_{1,d} + B_d, \quad b'_{i,d} = b_{i,d}\ (i \ge 2), \qquad \sum_i 2^i b'_{i,d} = 0.$$

It has the finite, exact decomposition

$$W = \prod_{d,i \ge 0} (E_{2^i d}^2 / E_{2^{i+1} d})^{c_{i,d}}, \qquad c_{i,d} = -\sum_{j>i} 2^{j-i-1} b'_{j,d}. \tag{7.15}$$

Indeed the coefficient of $E_{2^i d}$ on the right is $2c_{i,d} - c_{i-1,d}$, with $c_{-1,d} = 0$; this equals $b'_{i,d}$, including $i = 0$ by the zero weighted sum. All $c_{i,d}$ vanish for sufficiently large i. The product is one modulo two, so the norm condition makes it one modulo four. Equation (6.16) and independence by the least exponent make every $c_{i,d}$ even. Since $c_{0,d} \equiv b_{0,d} + b_{1,d}$ and $c_{i,d} \equiv b_{i+1,d}$ for $i \ge 1$, this proves

$$b_{0,d} \equiv b_{1,d} \pmod 2, \qquad b_{i,d} \equiv 0 \pmod 2 \quad (i \ge 2). \tag{7.16}$$

These are conditions on every odd divisor axis, including repeated prime powers in M.

The exact exponent norm from the upper to the lower level is

$$b_{0,d} = 3a_{0,d} + 2a_{1,d}, \quad b_{1,d} = -a_{0,d} + 2a_{2,d}, \quad b_{i,d} = 2a_{i+1,d} \quad (i \ge 2). \tag{7.17}$$

The complete integer preimage under (7.16) is

$$\begin{aligned} a_{0,d} &= -b_{1,d} + 2z_d, \qquad a_{1,d} = (b_{0,d} + 3b_{1,d})/2 - 3z_d, \\ a_{2,d} &= z_d, \qquad\qquad\quad a_{i+1,d} = b_{i,d}/2 \quad (i \ge 2). \end{aligned} \tag{7.18}$$

The kernel is generated on each axis by $(2, -3, 1, 0, \ldots)$. The following identities give the modularity conditions:

$$\sum_{j,d} a_{j,d} = \tfrac{1}{2} \sum_{i,d} b_{i,d} = 0, \tag{7.19}$$

$$\sum_{j,d} 2^j d\, a_{j,d} = \sum_{i,d} 2^i d\, b_{i,d}, \tag{7.20}$$

$$\sum_{j,d} v_p(d) a_{j,d} = \tfrac{1}{2} \sum_{i,d} v_p(d) b_{i,d} \quad (p \mid M). \tag{7.21}$$

Define integers

$$F = \sum_{i,d} \frac{2^s M}{2^i d} b_{i,d} = 24t, \qquad W_0 = \sum_d (M/d) b_{1,d},$$

$$H = \frac{\sum_{i,d} i\, b_{i,d} + \sum_d b_{1,d}}{2}, \qquad Z = \sum_d (M/d) z_d.$$

Integrality of H follows from the lower dyadic character condition and (7.16). Using weight zero in the first of the next two identities, direct substitution of (7.18) gives

$$\sum_{j,d} j\, a_{j,d} = H - \sum_d z_d, \tag{7.22}$$

$$T := \sum_{j,d} \frac{2^r M}{2^j d} a_{j,d} = F/2 - 3 \cdot 2^{r-3} W_0 + 3 \cdot 2^{r-2} Z. \tag{7.23}$$

For example the three leading source terms on one axis sum to $2^{r-2}(b_0-b_1)+3\cdot 2^{r-2}z$; the higher terms are $2^{r-i-2}b_i$. This proves (7.23) for every r. Thus the remaining upper conditions are precisely

$$\begin{aligned} S_p &:= \sum_{i,d} v_p(d) b_{i,d} \equiv 0 \pmod 4 \quad (p \mid M), \\ H - \sum_d z_d \equiv 0 \pmod 2, &\qquad 4t - 2^{r-3}W_0 + 2^{r-2}Z \equiv 0 \pmod 8. \end{aligned} \tag{7.24}$$

The factor three in (7.23) ensures the modulo-three part of the reciprocal Newman condition automatically.

Rational cusp tests give the remaining conditions. The data of g from (6.10)–(6.11) are

cusp denominator	width	ord(g)	positive leading amplitude
2^s	M	M	16
$2^{s-1},\ s\ge 3$	M	M	16
$2^{s-1},\ s=2$	M	$-M$	1/16
2	$2^{s-2}M$	$-2^{s-2}M$	1/16.

All these cusps are rational: their cusp conductors are one or two. To verify the orders, insert the exponents $8,-24,16$ at $\delta=1,2,4$ into $w\sum e_\delta \gcd(c,\delta)^2/(24\delta)$. For $4\mid c$ the parenthesized sum is $8-48+64=24$; for $c=2$ it is $8-48+16=-24$. For $4\mid c$ the eta amplitudes are all one, leaving the scalar 16; for $c=2$ the η_4^{16} contribution is $(2/4)^8=2^{-8}$, leaving 2^{-4}. The tame residue is independent of the distinction between analytic and rational parameters.

At $c=2^s$, the odd-prime valuation of the leading coefficient of u is $-S_p/2$. The lower character makes it integral. Equation (6.12) then has p-valuation $MS_p/2$, which must be even. Since M is odd, this gives every first condition of (7.24).

At $c=2^{s-1}$ the dyadic amplitude exponent of u is $-\sum_d b_{s,d}/2$; all lower dyadic axes contribute zero. For $s\ge 3$, the dyadic valuation of the tame residue is $4\,\mathrm{ord}_c(u)+M\sum_d b_{s,d}/2$. For $s=2$ both signs reverse, giving the same square condition. Consequently

$$\sum_d b_{s,d} \equiv 0 \pmod 4. \tag{7.25}$$

The remaining sign comes from $c=2$. The exact simplification of (6.10) is

$$\phi(2,\delta) = \frac{\delta}{48} + \frac{\gcd(2,\delta)^2}{48\delta} - \frac{1}{8}. \tag{7.26}$$

For even δ one has $C=1$, $s(d,C)=0$ and obtains this formula directly. For odd δ, $C=2$, $d=1$ and $s(1,2)=0$, giving the same expression. Put $w=2^{s-2}M$ and $v=\mathrm{ord}_{1/2}(u)$. Weight zero and the scalar $(-1)^A$ give the total leading phase of u as

$$A/2 + \sum_{i,d} b_{i,d}\phi(2,2^i d) = A + v/(2w).$$

For g, the phase is $8(-1/12)-24(-1/24)+16(-1/48)=0$, its order is $-w$, and its amplitude is $1/16$. The phase of the tame residue is therefore exactly

$$-wv/2 + wA + v/2 \equiv v(1-w)/2 \pmod{\mathbf{Z}}. \tag{7.27}$$

For $s\ge 3$ the width w is even. All amplitudes are positive; a rational square has positive sign, so v is even. Writing $B_0=\sum_d (M/d)b_{0,d}$, the exact cusp-order formula gives

$$v = t - 2^{s-5}B_0. \tag{7.28}$$

Indeed relative to $F/24$ the $i=0$ contribution changes from $2^sB_0/24$ to $2^{s-2}B_0/24$, a difference $-2^{s-5}B_0$; all $i\ge 1$ contributions agree.

We finish the integer elimination separately at the first two depths. The lower dyadic character and (7.16) make $\sum_d b_{1,d}$ even. The infinity congruence modulo four then gives $\sum_d d b_{0,d} \equiv 0 \pmod 4$. Since $d^{-1} \equiv d \pmod 4$,

$$4 \mid B_0, \qquad 2 \mid W_0. \tag{7.29}$$

With $z_d = 0$ for $d \neq M$, the last two conditions of (7.24) have a solution if and only if

$$\begin{array}{l|c} r \geq 5 & t \equiv 0 \pmod 2 \\ r = 4 & H \equiv W_0/2 - t \pmod 2 \\ r = 3 & H \equiv W_0/2 \pmod 2. \end{array} \tag{7.30}$$

For $r \geq 5$, the W_0 term is divisible by eight using (7.29), and the z_M term is divisible by eight; the equation is $4t = 0 \pmod 8$, while the first condition chooses $z_M \equiv H \pmod 2$. For $r = 4$, division of the displayed congruence by four after using W_0 even gives $z_M \equiv W_0/2 - t \pmod 2$. For $r = 3$, it gives $z_M \equiv W_0/2 - 2t \pmod 4$; compatibility with its prescribed parity is precisely the last row. These congruences determine a choice of z_M satisfying both conditions with $z_d = 0$ for $d \neq M$.

For $s \geq 4$, the correction in (7.28) is even by (7.29); thus v even makes t even. For $s = 3$, equations (7.16) and (7.25) give

$$H = \sum_d b_{1,d} + \sum_d b_{2,d} + \tfrac{3}{2} \sum_d b_{3,d} \equiv 0 \pmod 2.$$

The infinity Newman sum modulo eight, using $d^{-1} \equiv d \pmod 8$, gives $B_0 + 2W_0 \equiv 0 \pmod 8$. Equation (7.28) and v even yield $t \equiv B_0/4 \equiv W_0/2 \pmod 2$. This is the $r = 4$ row of (7.30).

For $s = 2$, one has $H = \sum_d b_{1,d} + \sum_d b_{2,d}$ even. Set $D_d = b_{0,d} + b_{1,d} + b_{2,d}$. Weight zero says $\sum_d D_d = 0$. For odd d,

$$(d-1)/2 \equiv \sum_{\substack{p \mid M \\ p \equiv 3 \ (4)}} v_p(d) \pmod 2.$$

Every S_p is divisible by four, so $\sum_d d D_d \equiv 0 \pmod 4$. Subtract this from the infinity Newman congruence modulo four: $\sum_d d b_{1,d} \equiv \sum_d d b_{2,d} \pmod 4$. Because each $b_{2,d}$ is even, multiplication by odd d does not change it modulo four, and (7.25) makes the right side zero. Hence $W_0 \equiv 0 \pmod 4$, which proves the $r = 3$ row.

Choose z_M as in the relevant row and set the others to zero. Then (7.18) satisfies every integral condition (6.8); it is a rational upper cusp unit. Its norm has exactly the exponents of u. Its infinity order equals A by (7.20), and its norm leading coefficient is $(-1)^A$. Multiplying by a restores the removed scalar $\lambda = (-1)^A a^2$. This proves the equality of functions, including constants, at all depths $r \geq 3$. □

7.5. **Invariant divisors and the two descents.** We first record invariant Picard descent for either cover $f : X \to Y$ at a deep level. Let x_0 be its rational ramified cusp and $y_0 = f(x_0)$. If a rational degree-zero line bundle L_0 is deck-invariant, normalize an isomorphism to its deck pullback to be the identity at x_0. The composite with its conjugate is a constant equal to one there, so it is a linearization. Its negative stabilizer signs form a reduced rational divisor of ramified cusps. Twisting by that divisor removes all inertia; the resulting bundle descends to a bundle on Y. Thus before degree correction it is $f^*E + C$. If $d = \deg E$, replace the pair by $E - dy_0$ and $C + 2dx_0$. Since $f^* y_0 = 2x_0$, this gives an equality of degree-zero classes

$$L_0 = f^* R + c, \qquad R \in J_Y(\mathbf{Q}), \quad c \in C_{\mathrm{rat}}(X). \tag{7.31}$$

If L_0 is torsion then c is torsion by Manin–Drinfeld, and $2R = f_*(L_0 - c)$ makes R torsion. Taking two-primary parts preserves (7.31).

Proof of Theorem 7.1. Assume $T_L = C_L$ and take $P \in T_N$. By (7.13), the point $v = (B-1)P$ is A-invariant. Apply (7.31) to α. The lower class is torsion, hence cuspidal by the induction hypothesis after taking its two-primary part. Consequently v is represented by a rational degree-zero cusp divisor E.

Choose a rational degree-zero divisor A_0 representing P. The rational cusp removes the Picard descent obstruction, so there is a $f \in \mathbf{Q}(X)^*$ with

$$(B-1)A_0 - E = \mathrm{Div}(f).$$

Taking the β norm gives $\mathrm{Div}(\mathrm{Nm}\, f) = -\beta_* E$. Thus $\mathrm{Nm}\, f$ is a lower cusp unit and a function norm. Lemma 7.2 supplies an upper cusp unit g with exactly the same norm. Hilbert 90 gives $f/g = B(t)/t$ for a rational function t. The divisor $A_0 - \mathrm{Div}(t)$ is therefore B-invariant away from the cusps.

The cover β is unramified away from cusps. Its invariant divisor there is the pullback of an integral rational divisor below. At a split closed point the coefficients are equal on both preimages; at an inert point the ramification index is one, so its coefficient descends unchanged. The same degree correction at the rational ramified infinity cusp gives

$$P = \beta^* R' + c', \qquad R' \in J_Y(\mathbf{Q}), \quad c' \in C_{\mathrm{rat}}(X).$$

Again R' is torsion because its double is the norm of a torsion point. Taking two-primary parts and using $T_L = C_L$ makes P rationally cuspidal. The base cases are Theorems 3.1, 5.1, and 6.1. The induction includes its first steps $n = 3$ and $n = 4$ by the separate integer calculations above. □

8. The prime three

Throughout this section, $K = \mathbf{Q}(\zeta)$, where $\zeta = e^{2\pi i/3}$, and τ is the nontrivial element of $\mathrm{Gal}(K/\mathbf{Q})$. A subscript 3 on a finite group denotes its 3-primary subgroup.

Theorem 8.1. *For every positive integer N,*

$$J_0(N)(\mathbf{Q})[3^\infty] = C_{\mathrm{rat}}(N)[3^\infty].$$

More precisely, for every $n \geq 2$ and every M prime to 3,

$$J_0(3^n M)(K)[3^\infty] = C_K(3^n M)_3 + \Sigma(3^n M)[3](K). \tag{8.1}$$

The sum in (8.1) *need not be direct.*

We first prove the two shallow cases, then establish unit norm equalities and the balanced-operator identity on the integral model. These results give (8.1) by induction.

8.1. Semistable levels.

Lemma 8.2. *Let $\mathscr{G}$ be a smooth commutative group over an unramified 3-adic DVR R_0. Over $R = R_0[\zeta]$, its formal 3-primary torsion is killed by 3, and τ acts on that torsion by -1. Over a DVR of absolute ramification index 4, the 3-primary torsion of any commutative formal group is likewise killed by 3, and there is no nonzero torsion of depth greater than 2.*

Proof. In integral parameters at the identity, the multiplication series has the form

$$[3](\boldsymbol{t}) = 3\big(\boldsymbol{t} + \text{terms of degree at least } 2\big) + \text{terms of degree at least } 3. \tag{8.2}$$

Indeed, modulo 3 its differential vanishes and it factors through relative Frobenius; in particular every coefficient of degree strictly between 1 and 3 is divisible by 3. If the minimum valuation of the coordinates is $r > e/2$, choose a coordinate attaining that minimum. Its linear term in $[3]\boldsymbol{t}$ has valuation $e + r$; the other terms have valuation at least $\min(e + 2r, 3r) > e + r$. Thus multiplication by 3 preserves nonzero points at this depth and increases their minimum depth by e. No such point can have 3-power order.

For $e = 2$, a point of positive depth has its triple at depth at least $3 > 1$, and therefore its triple is zero. Write $\lambda = \zeta - 1$. Since $\tau\lambda = -\zeta^2\lambda$, conjugation acts by -1 on the first formal quotient. For a formal torsion point P, $\tau P + P$ has depth at least 2, so it is zero by the first part of the argument. For $e = 4$, the triple of a point of depth at least 1 has depth at least $3 > 2$, and the same argument proves the last assertion. These are the particular cases of the formal bound in [8, Proposition 9] that we use. □

Lemma 8.3. *If $9 \nmid L$, then*

$$J_0(L)(K)[3^\infty] = J_0(L)(\mathbf{Q})[3^\infty] \oplus \Sigma(L)[3](K). \tag{8.3}$$

If $3 \nmid M$, the two degeneracy pullbacks give the same isomorphism

$$\Sigma(M)[3^\infty] \longrightarrow \Sigma(3M)[3^\infty]. \tag{8.4}$$

Proof. The Jacobian $A = J_0(L)$ is semistable at 3. After a finite unramified extension splitting the torus, Raynaud uniformization has a semiabelian group G and an unramified lattice Y, with

$$0 \longrightarrow G[3^a] \longrightarrow A[3^a] \longrightarrow Y/3^aY \longrightarrow 0.$$

Here we use the semistable uniformization in [8, Theorem 7]. For a point P with $\tau P = -P$, its image in $Y/3^aY$ is both fixed and negated, hence is zero. Its unique lift to $G[3^a]$ is K-rational and anti-invariant. The valuation quotient of the semiabelian group is a torus cocharacter lattice and is torsion free. The lift is therefore integral. Its reduction is again both fixed and negated, since the quadratic extension has unchanged residue field. It reduces to the identity and belongs to the formal group. Lemma 8.2 gives $3P = 0$.

Each nonzero cyclic subgroup in the anti-invariant part is a $\mathbf{Q}$-subgroup of type μ_3: its Galois action is the mod-3 cyclotomic character. Vatsal's theorem for odd multiplicative subgroups at a semistable prime puts it in $\Sigma(L)$ [51, Theorem 1.1]. Conversely the Galois action on the Shimura group is cyclotomic [28]. The two idempotents $(1 \pm \tau)/2$ give (8.3).

For (8.4), use the description of $\Sigma(L)$ by characters of $(\mathbf{Z}/L\mathbf{Z})^*$ trivial on -1, on the elliptic stabilizer relations $d^2 + 1 = 0$ and $d^2 + d + 1 = 0$, and on the parabolic relations $(d-1)^2 = 0$; see [28, Theorems 1 and 4]. A 3-primary character at level $3M$ ignores $(\mathbf{Z}/3\mathbf{Z})^*$. The relations of order dividing 4 impose nothing on it. Every square-zero parabolic residue at level M lifts with $d = 1 \pmod 3$, as does every cubic elliptic residue, since $d = 1$ is the unique cubic elliptic root modulo 3. Thus the remaining relations at the two levels are identical. The matrices inducing the two degeneracies preserve the lower-right entry d; they consequently give the same pullback of characters. This proves the isomorphism, including even M. At the genus-zero small levels the assertion is vacuous. □

Proposition 8.4. *If $9 \nmid N$, then*

$$J_0(N)(\mathbf{Q})[3^\infty] = C_{\mathrm{rat}}(N)_3.$$

Proof. By Theorem 2.1, it remains to consider

$$N = 3\prod_r r^{a_r}, \qquad r \equiv 1 \pmod 3.$$

Here $\psi(N) = 4\prod_r r^{a_r-1}(r+1)$ is a 3-adic unit and $5 \nmid N$. At level $5N$, Yoo's theorem applies.

Write $\alpha, \beta : X_0(5N) \to X_0(N)$ for $z \mapsto z, 5z$ and $d_- = \alpha^* - \beta^*$. Ribet's degeneracy-kernel theorem identifies the kernel of $(x, y) \mapsto \alpha^* x + \beta^* y$ with the antidiagonal Shimura subgroup [42, §2, Proposition 1]. Thus $\ker(d_-)$ is contained in $\Sigma(N)$. Its rational odd part is zero, so d_- is injective on rational 3-primary torsion. The identities $w_5\alpha^* = \beta^*$ and $w_5\beta^* = \alpha^*$ give $w_5 d_- = -d_-$.

Let P_c be the sum of the cusps of denominator $c \mid N$. Its degree is $\varphi(\gcd(c, N/c))$, and its cusp width is $a_c = N/\gcd(c^2, N)$. The two corresponding upper packets have the same residue

degree. Directly from their widths,

$$\alpha^* v = (5v, v), \qquad \beta^* v = (v, 5v), \qquad w_5(v_0, v_1) = (v_1, v_0), \qquad d_- v = 4(v, -v)$$

on the rational cusp lattice. Define the rational cuspidal divisor $A_N = \sum_{c|N} a_c P_c$. It has degree $\psi(N)$, and the orders of the two discriminants at the upper packets give

$$\operatorname{Div} \frac{\Delta(z)}{\Delta(5z)} = 4(A_N, -A_N). \tag{8.5}$$

In the cusp lattice tensored with $\mathbf{Z}_3$, every minus class has a representative $(v, -v)$. Replace v by

$$v_0 = v - \frac{\deg v}{\psi(N)} A_N.$$

Then v_0 has degree zero, and (8.5) says that this replacement does not change the class. Since 4 is a unit, the minus 3-primary cuspidal group upstairs equals the image under d_- of the lower rational cuspidal group. These lattice operations give integral divisor representatives: use the integral degree-zero basis $P_c - (\deg P_c)\infty$, omitting $P_c = \infty$, for the lower cusp lattice, and reduce its $\mathbf{Z}_3$-coefficients modulo a common exponent of the finite cuspidal class group. This preserves degree zero exactly. Applying this to $d_- P$ and then using injectivity proves the proposition. □

8.2. **The cubic quotient at level $9M$.** Put $Y = X_0(M)$, $3 \nmid M$. Conjugation by $\operatorname{diag}(3, 1)$ identifies the group for $X = X_0(9M)$ with $\pm\Gamma(3) \cap \Gamma_0(M)$. The geometric deck group over Y is $\mathrm{PSL}_2(\mathbf{F}_3) = A_4$. Denote its normal four-group by V. The quotient $Z = X/V$ has the rational model

$$f : Z \longrightarrow Y, \qquad \mathbf{Q}(Z) = \mathbf{Q}(Y)(w), \qquad w^3 = j. \tag{8.6}$$

Put

$$E_4(z) = 1 + 240 \sum_{m \geq 1} \left(\sum_{d|m} d^3 \right) q^m, \qquad q = e^{2\pi i z},$$

and set $\gamma_2 = E_4/\eta^8$. For the standard generators

$$S_0 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad T_0 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

of $\mathrm{SL}_2(\mathbf{Z})$, the transformation laws give $\gamma_2(S_0 z) = \gamma_2(z)$ and $\gamma_2(T_0 z) = \zeta^{-1}\gamma_2(z)$. Also $\gamma_2^3 = E_4^3/\Delta = j$. This cubic character is the character of A_4/V. Moreover

$$\gamma_2(3z) = q^{-1} \frac{E_4(3z)}{\prod_{m \geq 1}(1 - q^{3m})^8}$$

has rational coefficients. This expansion identifies the $\mathbf{Q}$-model in (8.6); compare [32, §4].

At every cusp of Y, the pole order of j divides M and is prime to 3. Thus f is geometrically connected and totally ramified at every cusp, with unchanged residue field. Pullback on Pic^0 is injective: a nontrivial kernel would give a nontrivial connected étale factor of the degree-three cover, forcing that cover itself to be étale. Dually its norm kernel is connected. Write

$$P_f = \ker(\mathrm{Nm}_f : \mathrm{Jac}(Z) \to \mathrm{Jac}(Y)).$$

Proposition 8.5. *For every finite unramified extension $F/\mathbf{Q}_3$,*

$$3P_f(F)[3^\infty] = 0.$$

There is a fine compact model after a tame quartic extension whose whole Jacobian has central inertia weight 1 *and elliptic-tail cotangent weight* i.

Proof. On the Legendre line $X(2)$ with coordinate ℓ, the exact equations are

$$\begin{aligned} j &= \frac{256(\ell^2-\ell+1)^3}{\ell^2(1-\ell)^2}, \\ y &= 2\ell-1, \qquad x^3 = 4\ell(\ell-1), \qquad y^2 = x^3+1, \\ w &= \frac{16(\ell^2-\ell+1)}{x^2}, \qquad x = \frac{\ell(\ell-1)w}{4(\ell^2-\ell+1)}. \end{aligned} \tag{8.7}$$

Cubing the last two expressions verifies both directions of the function-field identification. On the modular full-six curve $r^2 = h^3+1$, the map to the Legendre line is

$$\ell_K = \frac{(r+1)(r-3)^3}{(r-1)(r+3)^3} \qquad [26,\ §7.1].$$

The doubling formula on this elliptic curve is

$$x(2Q) = \frac{h^4-8h}{4(h^3+1)}, \qquad y(2Q) = \frac{r^4+18r^2-27}{8r^3}.$$

Substitution and factorization give

$$\frac{y(2Q)+1}{2} = \frac{1}{1-\ell_K}. \tag{8.8}$$

The change $\ell_K \mapsto 1/(1-\ell_K)$ changes the Legendre labeling. Therefore the kernel of this quotient map is the four translations by the two-torsion, which is precisely the subgroup V.

Enlarge F unramified to contain i and the required tame marking labels. Let B be a geometrically connected fine curve with symplectic full-four marking and cyclic M-marking. When the markings overlap, choose a component inside the fixed-pairing locus and use its stabilizer. The marking levels 4 and M are prime to 3, so these marking problems are finite étale on the elliptic locus. The full-four marking removes automorphisms. The fine curve has smooth good reduction, including its tame Tate cusp charts [25, 4.7.1–4.7.2, 8.6.8].

The map $B \to X(2)$ is étale on the elliptic locus in characteristic 3. At the supersingular curve $y^2 = x^3 - x$, all origin-preserving automorphisms have the form

$$(x,y) \longmapsto (u^2x+a, u^3y), \qquad u^4 = 1, \quad a \in \mathbf{F}_3.$$

Their action on the three nonzero two-torsion points is the affine permutation group, and its kernel consists exactly of ± 1. Full-four marking removes that kernel. Further prime-to-three cyclic marking is étale. This proves the assertion also when M is even or has repeated prime factors.

Normalize $Z \times_Y B$ and denote the resulting curve by Z^f. By (8.7) it is the normalization of the same cubic elliptic cover pulled back to B. Put $L = F(\pi)$, $\pi^4 = 3$. Write k for the common residue field of F, L, and B_k for the special fiber of the smooth model of B. Make the substitutions

$$x = -1 + \pi^2 u, \qquad y = \pi^3 v.$$

The resulting proper Weierstrass model is

$$v^2 = u^3 - \pi^2 u^2 + u. \tag{8.9}$$

Its special fiber $E_0 : v^2 = u^3 + u$ is smooth, including infinity: the affine derivative in u is -1. The map to the Legendre line is $\ell = (1+\pi^3 v)/2$. Its only point of indeterminacy is infinity on the special fiber. In the exact parameters

$$t = -u/v, \qquad z = -1/v$$

the equation becomes

$$z = t^3 - \pi^2 t^2 z + t z^2 = t^3 U(t,\pi), \qquad U(0,\pi) = 1, \qquad \ell = \frac{1-(\pi/t)^3/U}{2}. \tag{8.10}$$

Blow up the ideal (π, t). On the t-chart put $\pi = ta$; the equation $\pi = ta$ is the usual semistable node, and the last expression in (8.10) is regular. On the other chart put $t = \pi b$; then

$$\ell = \frac{b^3U - 1}{2b^3U}, \qquad \ell^{-1} = \frac{2b^3U}{b^3U - 1},$$

which is regular at $b = 0$. These two charts resolve the whole map. The strict elliptic component maps to $\ell = 1/2$; the exceptional projective line maps by

$$\ell = (1 - a^3)/2. \tag{8.11}$$

The two components meet once.

Pull back along $B \to X(2)$ and normalize. Its étaleness on the elliptic locus gives one copy of E_0 and one ordinary node over every supersingular point of B. The central function field is obtained from $k(B_k)$ by adjoining $\ell^{1/3}$. Since $k(B_k)/k(\ell)$ is separable, $d\ell \neq 0$, so ℓ is not a cube. A one-variable function field over the perfect field k has a unique purely inseparable extension of degree 3. The central component is therefore one connected Frobenius twist of B_k.

It remains to check every cusp of this compactification. In its fine chart, $q = \xi s^a$, where $3 \nmid a$ and ξ has order prime to 3. The identity

$$w = \frac{1 + 240 \sum_{m \geq 1} (\sum_{d|m} d^3) q^m}{q^{1/3} \prod_{m \geq 1} (1 - q^m)^8}$$

shows that the extension is exactly the extension obtained by adjoining $q^{1/3}$. Adjoin $\xi^{1/3}$ unramified and choose integers b, c with $ba + 3c = 1$. Then

$$t_0 = s^c \big(q^{1/3}/\xi^{1/3}\big)^b, \qquad t_0^3 = s.$$

The integral closure is $\mathcal{O}_L[[t_0]]$: this regular ring is finite over $\mathcal{O}_L[[s]]$ and has the required fraction field. Thus there is no further vertical cusp component. These charts glue with those on the elliptic locus because they normalize the same generic field. The whole special fiber has one central component and one elliptic tail at each supersingular point, attached once. Its dual graph is a tree, so the whole fine Jacobian has good reduction over L.

Let $\sigma(\pi) = i\pi$, with σ fixing F. The exact semilinear actions are

$$\sigma(u, v) = (-u, iv), \qquad \sigma(t) = it, \qquad \sigma(a) = a. \tag{8.12}$$

Hence inertia is the identity on the central component and its square is $[-1]$ on each tail. For a prime $\ell_0 \neq 3$, the rational Tate module of the compact-type Jacobian is the direct sum of the central and tail Tate modules. Norm to $\mathrm{Jac}(B)$ is an isomorphism on the central rational Tate module, because (8.11) is purely inseparable of degree 3, and is zero on the tails. Consequently the rational Tate module of the connected fine Prym P_f^f is exactly the tail summand. Its unramifiedness over L gives good reduction of P_f^f itself by the Néron–Ogg–Shafarevich criterion [2]. Faithfulness of the rational Tate representation of an abelian variety now gives

$$\sigma^2 + 1 = 0 \quad \text{on the special fiber of } P_f^f. \tag{8.13}$$

If $Q \in P_f^f(F)[3^\infty]$, its reduction is σ-fixed. By (8.13) it is killed by 2, hence is zero. It belongs to the Prym's formal group over L. Lemma 8.2 gives $3Q = 0$. Here the ramification index is exactly 4; indeed $\zeta = (-1 + i\pi^2)/2$ already belongs to L.

Finally we descend the assertion from the fine curve. For odd M, the full-four geometric deck group is $\mathrm{PSL}_2(\mathbf{Z}/4\mathbf{Z}) \simeq S_4$. Its abelianization is C_2. For even M, the relevant group is contained in the inverse image of the mod-two Borel and is a 2-group. In the odd case the full-four and cubic covers are linearly disjoint, since S_4 has no quotient C_3; in the even case their degrees are coprime. Thus these are also the geometric deck groups of $Z^f \to Z$.

For a proper connected Galois cover, the kernel of Picard pullback injects into its character group: a trivialization of the pulled-back bundle has deck translates differing by constants, and

the resulting character is trivial only when the trivialization descends. The displayed groups have no nontrivial 3-primary characters. Hence $\mathrm{Jac}(Z)[3^\infty] \to \mathrm{Jac}(Z^f)[3^\infty]$ is injective. Norm and pullback commute on the normalized square, as is seen on divisors away from its branch locus. The connected coarse Prym therefore maps into the connected fine Prym. The proved exponent descends by this injection.

For later use, the same construction identifies the whole special cotangent representation: it is 1 on the central factor and i on the tails, since $\sigma(du/v) = i\,du/v$. □

8.3. **The base case at $9M$.**

Lemma 8.6. *Let k contain μ_3, and let $f : Z \to Y$ be a morphism of smooth proper geometrically connected curves. Suppose that ρ is an automorphism of Z over Y, fixing $R_0, R_1 \in Z(k)$, with images y_0, y_1. If $L \in \mathrm{Pic}(Z)[3]$, $s \in \mathrm{Pic}(Y)[3]$, and*

$$(1-\rho^*)L = f^*s,$$

the μ_3-cover associated to s and split at y_0 is split at y_1. For $Y = X_0(M)$, $3 \nmid M$, a nonzero $s \in \Sigma(M)[3](K)$ cannot have split fibers at both 0 and ∞.

Proof. The exact sequence

$$0 \longrightarrow k^*/k^{*3} \longrightarrow H^1(Z, \mu_3) \longrightarrow \mathrm{Pic}(Z)[3] \longrightarrow 0$$

is split by evaluation at R_0. Let h_L be the unique lift of L whose value there is zero, and normalize h_s at y_0. Since $\rho R_0 = R_0$, the classes $(1-\rho^*)h_L$ and f^*h_s have the same Picard image and the same value at R_0. They are equal. At R_1 the first evaluates to zero, so the second does also.

For the last assertion use the point-generator convention for $X_1(M)$: its points are pairs (E, P), and a diamond operator sends P to aP. The M-gon cusp above 0 has the generator $P = (1, 1)$, so its diamond fiber is split over $\mathbf{Q}$. At the one-gon cusp above ∞ the generator is ζ_M, and Galois acts on the diamond fiber through its cyclotomic character modulo M. The widths at these endpoints agree upstairs and downstairs. The maximal compact étale quotient of this diamond cover has character group $\Sigma(M)$ [28]. Thus, for a character χ representing s, its 0-based fiber at ∞ is the character

$$\mathrm{Gal}(\overline{\mathbf{Q}}/K) \longrightarrow (\mathbf{Z}/M\mathbf{Z})^* \xrightarrow{\chi} \mu_3.$$

The first map is onto because $K \cap \mathbf{Q}(\zeta_M) = \mathbf{Q}$. This value vanishes precisely when χ, and hence s, is zero. □

Proposition 8.7. *For every M prime to 3,*

$$J_0(9M)(\mathbf{Q})[3^\infty] = C_{\mathrm{rat}}(9M)_3.$$

Proof. First suppose $X_0(M)$ has no elliptic point of order 3. In (8.6) the zeros of j away from such elliptic points have orders divisible by 3. Thus the branch locus of f consists only of cusps. Let $\rho(w) = \zeta w$, and set $\delta = 1 - \rho^*$. On P_f the identity $1 + \rho^* + \rho^{*2} = 0$ gives

$$\delta^2 = -3\rho^*.$$

For $P \in P_f(\mathbf{Q})[3^\infty]$, Proposition 8.5 gives $3P = 0$. Consequently δP is ρ-invariant and

$$\tau(\delta P) = (1 - \rho^{*-1})P = -\rho^{*-1}\delta P = -\delta P.$$

Represent δP by a line bundle and normalize its ρ-linearization at the rational ramified cusp above 0. Its threefold product is then the identity. The character at any branch point has an exponent in $\mathbf{F}_3$. Choose an automorphism of $\overline{\mathbf{Q}}$ that fixes $\mathbf{Q}(\zeta_M)$ and restricts to τ on K. It fixes every branch cusp. Its action on a character exponent is untwisted, because it inverts both

ρ and ζ. Anti-invariance negates the same exponent. Every exponent is therefore zero. The linearized bundle descends without a divisor correction:

$$\delta P = f^* s, \qquad s \in J_0(M)(K)[3], \qquad \tau s = -s.$$

Here pullback is injective, so both the torsion and the anti-invariance descend. Lemma 8.3 puts s in $\Sigma(M)[3](K)$; Lemma 8.6 then gives $s = 0$. We have proved $\rho^* P = P$.

Normalize a linearization of P at the same cusp. Its branch characters are Galois stable. At a ramified point the tangent character generates the character group; twisting by an integral multiple of that point cancels any specified exponent. We can therefore cancel all characters by a rational cuspidal divisor D. The remaining bundle descends to Y. Its degree equation says that $\deg D$ is a multiple of 3. Subtract $(\deg D)$ times the rational ramified cusp from D, and make the corresponding degree correction downstairs. This does not change the inertia characters, since the subtracted coefficient is divisible by 3. It gives

$$P = f^* R + [D], \qquad \deg R = \deg D = 0, \qquad R \in J_0(M)(\mathbf{Q}).$$

Taking norm and using $\operatorname{Nm} P = 0$ gives $3R = -f_*[D]$. The right side is cuspidal torsion, so R is torsion. After simultaneous primary projection, Proposition 8.4 makes R rational cuspidal. Hence P is rational cuspidal on Z.

This also treats the whole rational Jacobian of Z. For $Q \in \mathrm{Jac}(Z)(\mathbf{Q})[3^\infty]$, its norm is a rational cusp class on Y. Every cusp divisor downstairs lifts to a cusp divisor upstairs with the same degree and pushforward, since the cusp residue fields do not change. Subtract the 3-primary part of such a lift. The remainder lies in the connected norm kernel P_f, just treated. Cuspidal classes on Z are torsion: pull back to $X_0(9M)$, apply Manin–Drinfeld, and push down. Thus the primary projection used here is legitimate.

We next recover the whole A_4-cover, keeping the rational structure of its intermediate maps. From a cyclic nine-isogeny take its middle elliptic curve. Its dual incoming line and its outgoing line are two distinct lines in the three-torsion. Over K, the pairing determines all four lines. They define four maps

$$f_0, f_\infty, f_+, f_- : X_0(9M)_K \longrightarrow X_0(3M)_K.$$

The first two are defined over $\mathbf{Q}$; conjugation exchanges f_+ and f_-. Each has the same composite π with the forgetful map $a : X_0(3M) \to X_0(M)$. On a dense unramified open, the fiber-product correspondence gives

$$\sum_i f_i^* = \pi^* a_*. \tag{8.14}$$

The equality extends across the finite branch locus as an equality of the closed correspondences.

For a subgroup H, write $N_H = \sum_{h \in H} h$. The four order-three subgroups of A_4 have disjoint nonidentity elements, and their eight nonidentity elements are exactly those outside V. Coefficient by coefficient this proves

$$\sum_{i=1}^{4} N_{C_{3,i}} = N_{A_4} - N_V + 4. \tag{8.15}$$

Put $e = 1 - N_V/4$. It is defined over $\mathbf{Q}$, since Galois preserves V, and $e\pi^* = 0$. For a rational point $P = eP$, let $q_i = f_{i,*}P$. Then q_0, q_∞ are rational and $q_- = \tau q_+$. Set

$$\bar{q} = (q_+ + q_-)/2, \qquad s = (q_+ - q_-)/2.$$

By Lemma 8.3, $s \in \Sigma(3M)[3](K)$, and (8.4) writes $s = a^* s_0$. Thus $f_i^* s = \pi^* s_0$ for every i. Combining (8.14) and (8.15) gives the exact formula

$$4P = e f_0^*(q_0 - \bar{q}) + e f_\infty^*(q_\infty - \bar{q}). \tag{8.16}$$

Both lower points on the right are rational torsion, hence rational cusp classes by Proposition 8.4. The maps preserve cusps. The operator N_V also takes rational cusp divisors to rational cusp divisors, since Galois permutes its four summands. It follows that the non-V part is rational

cuspidal. The V-part is pulled back from $\mathrm{Jac}(Z)$ by norm divided by 4, so it is rational cuspidal as well.

Suppose now that elliptic points of order 3 exist on $X_0(M)$. A root of d^2+d+1 modulo M then forces every prime dividing M to be $1 \pmod 3$; $M=1$ is allowed. In particular $5 \nmid M$, and $X_0(5M)$ has no such elliptic point. The result just proved applies at $45M$. Set $N=9M$ and use the rational minus map d_- at the auxiliary prime 5. Its injectivity and cusp-packet formulas are as in the proof of Proposition 8.4. This time the degree relation is supplied by

$$F(z)=\eta(3z)^8.$$

At level 9 the two Newman sums are 24, its weight is 4, and its squareclass is 3^8, a square. It is a modular form with trivial character, nonzero in the upper half-plane. Its cusp divisor A at level N has degree

$$\deg A=\psi(N)/3=4M\prod_{r\mid M}(1+1/r),$$

a 3-adic unit, and direct cusp orders give

$$\mathrm{Div}\,\frac{F(z)}{F(5z)}=4(A,-A).$$

The same degree elimination as in (8.5) yields $C_{\mathrm{rat}}(5N)_3^{w_5=-1}=d_-C_{\mathrm{rat}}(N)_3$. For a rational torsion point, the equality at $5N$ and injectivity of d_- now give the result at N. □

Proposition 8.8. *For every M prime to 3,*

$$J_0(9M)(K)[3^\infty]^{\tau=-1}\subseteq C_K(9M)_3+\Sigma(9M)[3](K).$$

Proof. We first analyze an anti-invariant point on the whole $\mathrm{Jac}(Z)$ of (8.6). Pull it back to the good fine Jacobian constructed in Proposition 8.5. The generator $\sigma(\pi)=i\pi$ restricts to τ, since $\zeta=(-1+i\pi^2)/2$. If $\tau Q=-Q$, then $\sigma Q=-Q$ and $\sigma^2Q=Q$. On the central special factor $\sigma=1$, and on each tail $\sigma^2=[-1]$. In both cases the reduction of Q is killed by 2. It is 3-primary, so it is zero.

The special cotangent weights on the whole good Jacobian are 1 and i. Lift a basis vector of weight χ to the formal augmentation ideal and replace its lift by

$$\frac{1}{4}\sum_{j=0}^{3}\chi^{-j}\sigma^j f.$$

This produces an exact weight vector: σ fixes i and the residue field. These vectors are integral parameters by formal smoothness. Replace each by $(f-[-1]^*f)/2$ to make it odd. The weights and the leading cotangent vectors are unchanged. For such a parameter,

$$\sigma(f(Q))=(\sigma f)(\sigma Q)=-\chi f(Q). \tag{8.17}$$

If $f(Q)\neq 0$ has π-valuation r, comparison of leading residues gives $i^r=-\chi$. Thus the central depths are $2 \pmod 4$, the tail depths are $3 \pmod 4$, and every coordinate has depth at least 2.

The cubic deck action in (8.9) is exactly

$$\rho(u)=\zeta u+\frac{\pi^2-i}{2},\qquad \rho(v)=v. \tag{8.18}$$

On a special tail it is $u\mapsto u+i$; it preserves du/v. It is the identity on the central component. Therefore every linear coefficient of $\delta=1-\rho^*$ on the whole formal Jacobian is divisible by π. For Q of depth at least 2, the linear terms of δQ have depth at least 3 and the nonlinear terms have depth at least 4. Lemma 8.2 gives $\delta Q=0$. Equation (8.2) also gives $[3]Q$ depth at least $\min(4+2,3\cdot 2)=6$, so $3Q=0$. The fine pullback is injective on 3-primary torsion, as proved above. Both conclusions hold on the original $\mathrm{Jac}(Z)$.

When the cubic cover has no noncuspidal branch point, normalize the linearization of this invariant anti point at the rational ramified cusp. Choose conjugation fixing $\mathbf{Q}(\zeta_M)$, as in the

preceding proof. The branch exponents are both fixed and negated, hence zero. The point is therefore the pullback of an anti point in $J_0(M)(K)[3]$, which is in $\Sigma(M)[3]$ by Lemma 8.3.

For the A_4 complement, if $P = eP$ is anti-invariant, then $q_- = -\tau q_+$, whereas q_0, q_∞ are anti-invariant. Put

$$s = (q_+ + q_-)/2, \qquad r = (q_+ - q_-)/2.$$

Here s is anti-invariant and r is rational. All anti terms are common pullbacks from $\Sigma(M)[3]$ and are annihilated by e. Equation (8.15) consequently gives

$$4P = e(f_+^* - f_-^*)r. \tag{8.19}$$

By the shallow rational theorem, r is represented by a rational cusp divisor. Thus this complement is represented by K-cusp divisors. The V-part just considered gives the Shimura summand. Its pullback is the inflation of the bottom-right matrix character, also for the middle degeneracy $z \mapsto 3z$. This proves the assertion when the branch locus is cuspidal.

For the remaining M, add the auxiliary prime 5 and set $N = 9M$. We use

$$d_+ = \alpha^* + \beta^* : J_0(N) \longrightarrow J_0(5N).$$

Ribet's antidiagonal-kernel theorem gives injectivity on the entire geometric 3-primary group: $d_+P = 0$ implies $(P, P) = (s, -s)$, hence $2P = 0$. The two pullbacks identify $\Sigma(N)_3$ with $\Sigma(5N)_3$ and agree there. Indeed a 3-primary character ignores the order-four factor at 5; there are no cubic elliptic relations at either level because $9 \mid N$; the parabolic conditions lift with residue 1 at 5. The matrix maps preserve d. Thus $d_+ = 2\alpha^*$ is an isomorphism on these Shimura groups.

For a lower anti point Q, the proved upper case gives $d_+Q = c + s$. Absorb $s = d_+s_0$ with $s_0 \in \Sigma(N)[3](K)$. We have

$$d_+(Q - s_0) = c, \qquad \tau c = -c, \qquad w_5 c = c.$$

It remains to descend this cusp class. Let L_N be the free $\mathbf{Z}_3$-module on K-closed cusps. Label the two upper packets by their α-images. The widths and the numerator permutation give

$$\alpha^* v = (5v, v), \qquad \beta^* v = (\tau v, 5\tau v), \qquad w_5(U, V) = (\tau V, \tau U). \tag{8.20}$$

To check the permutation, the cusp $a/(5d)$ at the lower level is equivalent to a cusp a'/d with $a' = 5a \pmod{\gcd(d, N/d)}$. The other degeneracy sends a/d to $5a/d$. Pullback uses the inverse permutation on one packet. On K-closed cusp orbits, the cyclotomic permutation of exponent 5 is exactly $\tau = \tau^{-1}$: all its other cyclotomic coordinates lie in the subgroup fixing K.

On the anti block, (8.20) therefore gives

$$d_+ v = 4(v, -v), \qquad d_- v = 6(v, v), \qquad w_5(U, V) = (-V, -U). \tag{8.21}$$

Apply the commuting projectors $(1 - \tau)/2$ and $(1 + w_5)/2$ to a degree-zero cusp representative of c. It now lies in precisely the block $4(v, -v)$. An anti divisor v has degree zero, because degree is conjugation invariant and 2 is a unit. Division by 4 in the finite class group, followed by reduction modulo its exponent of the coefficients in the degree-zero basis $c - (\deg c)\infty$, $c \neq \infty$, of K-closed cusps, produces an integral lower K-cusp divisor of degree zero with class c_0 and $d_+c_0 = c$. Injectivity gives $Q = s_0 + c_0$, as required. □

Corollary 8.9. *Equation* (8.1) *holds for* $n = 2$.

Proof. Decompose a finite 3-primary point by $(1 \pm \tau)/2$. Proposition 8.7 treats its rational part and Proposition 8.8 treats its anti part. The reverse inclusion follows from Manin–Drinfeld and the definition of Σ. □

8.4. **Rational unit norms.** For a curve $X_0(L)$, let $\mathcal{U}_L(F)$ be the group of F-rational functions with divisors supported on cusps; constants are included. For this subsection put

$$N = 3^r M, \quad L = 3^{r-1} M, \quad r \geq 2, \quad 3 \nmid M, \qquad \beta : X_0(N) \to X_0(L), \quad z \mapsto 3z.$$

Proposition 8.10. *For every such N,*

$$\mathcal{U}_L(\mathbf{Q}) \cap \mathrm{Nm}_\beta\, \mathbf{Q}(X_0(N))^* = \mathrm{Nm}_\beta\, \mathcal{U}_N(\mathbf{Q}). \tag{8.22}$$

Proof. Write $E(x) = \prod_{m \geq 1}(1 - x^m)$. The integral eta-product theorem [46, Theorem 7, Corollary 8, Lemmas 15–19] implies that every rational modular unit is a rational constant times an eta product with integer exponents. To apply it, multiply the unit by a power of Δ to remove all cusp poles, clear its bounded denominators, and undo that power after applying the theorem. Thus

$$u = c \prod_{\substack{d \mid M \\ 0 \leq i \leq r-1}} \eta(3^i dz)^{b_{i,d}}, \qquad b_{i,d} \in \mathbf{Z}. \tag{8.23}$$

The weight-zero eta lattice consists of

$$\begin{gathered} \sum b_{i,d} = 0, \qquad \sum 3^i d\, b_{i,d} \equiv 0 \pmod{24}, \\ \sum \frac{L}{3^i d}\, b_{i,d} \equiv 0 \pmod{24}, \qquad \sum v_\ell(3^i d) b_{i,d} \equiv 0 \pmod 2 \quad \text{for every prime } \ell. \end{gathered} \tag{8.24}$$

These are the two eta multiplier congruences and the trivial square character. They also fix all character conditions in the source lattice below.

To complete the modular cover at a vertical valuation, let

$$R = \widehat{(\mathbf{Z}_3[[q]])}_{(3)}, \qquad F = \mathrm{Frac}(R), \qquad t^3 = q, \qquad S = R[t].$$

The polynomial $T^3 - q$ has irreducible reduction over $\mathbf{F}_3((q))$. The ring S is a DVR with uniformizer 3, residue field $\mathbf{F}_3((t))$, and residue degree 3. It is the full integral closure of R. The function-field identity $\mathbf{Q}(X_0(m)) = \mathbf{Q}(j(z), j(mz))$, together with grouping series by exponents modulo 3, embeds the upper field into $F(t)$, compatibly with $\beta : q = t^3$. The coefficient of t^2 in the basis expansion of $j(t)$ contains q^{-1} and only nonnegative additional powers of q, so $j(t) \notin F$. The modular cover has degree 3, so its base change is $F(t)/F$, and its function norms become determinant norms there.

The trace of S is contained in $3R$. If a norm W lies in $1 + 3R$, its witness has valuation zero, because the norm multiplies vertical valuations by 3. Its residue has cube 1, so it is 1. Write the witness as $1 + 3z$; determinant expansion gives

$$W = \mathrm{Nm}(1 + 3z) \equiv 1 + 3 \operatorname{Tr} z \equiv 1 \pmod 9. \tag{8.25}$$

Let $A_\infty = \sum 3^i d\, b_{i,d}/24$. At the totally ramified rational cusp, $q = t^3$, a function norm with order A_∞ has leading coefficient a^3, $a \in \mathbf{Q}^*$. Divide u by this constant norm once. All products in the remainder of the proof are leading-normalized. For each $d \mid M$, put $B_d = \sum_i 3^i b_{i,d}$ and form the local element

$$V(t) = t^{A_\infty} \prod_{d \mid M} E(t^d)^{B_d} \in F(t).$$

The Euler norm identities are

$$\mathrm{Nm}\, E(t^d) = \frac{E(q^d)^4}{E(q^{3d})} \quad (3 \nmid d), \qquad \mathrm{Nm}\, E(t^{3m}) = E(q^m)^3, \qquad \mathrm{Nm}\, t = q. \tag{8.26}$$

They follow by multiplying the three factors at $t, \zeta t, \zeta^2 t$, separating the Euler factors with index divisible by 3. The exponent vector of $W = u/\mathrm{Nm}\, V$ is

$$b'_{0,d} = b_{0,d} - 4B_d, \qquad b'_{1,d} = b_{1,d} + B_d, \qquad b'_{i,d} = b_{i,d}\ (i \geq 2), \qquad \sum_i 3^i b'_{i,d} = 0.$$

Hence

$$W = \prod_{d,i\geq 0} \left(\frac{E(q^{3^i d})^3}{E(q^{3^{i+1} d})} \right)^{c_{i,d}}, \qquad c_{i,d} = -\sum_{j>i} 3^{j-i-1} b'_{j,d}.$$

Each parenthesis is 1 (mod 3), so (8.25) applies. For $R_m = E(q^m)^3/E(q^{3m})$, the series $(R_m - 1)/3$ (mod 3) starts with $-q^m$. Choosing the least index with a nonzero coefficient shows that every $c_{i,d}$ is divisible by 3. This proves exactly

$$b_{0,d} + b_{1,d} \equiv 0 \pmod 3, \qquad b_{i,d} \equiv 0 \pmod 3 \quad (i \geq 2). \tag{8.27}$$

Coefficient divisibility follows from vertical divisibility because $9R \cap \mathbf{Z}_3[[q]] = 9\mathbf{Z}_3[[q]]$.

The norm map on eta exponents is the full stencil

$$b_0 = 4a_0 + 3a_1, \qquad b_1 = -a_0 + 3a_2, \qquad b_i = 3a_{i+1} \quad (i \geq 2) \tag{8.28}$$

on each axis d. An integral solution is

$$a_{0,d} = -b_{1,d}, \quad a_{1,d} = (b_{0,d} + 4b_{1,d})/3, \quad a_{2,d} = 0, \quad a_{i+1,d} = b_{i,d}/3. \tag{8.29}$$

All integral solutions add $z_d(3, -4, 1, 0, \ldots)$. The source weight is one third of the target weight; the infinity sum is unchanged. For $\ell \neq 3$, the target character moment is three times the source moment. For $\ell = 3$,

$$\sum i b_{i,d} = 3 \sum j a_{j,d} - 3 \sum a_{j,d} + 2a_{0,d}.$$

Thus every parity condition in (8.24) passes to the source. The kernel vector changes the last moment by $-2z_d$. Only the reciprocal sum remains.

Set

$$F_0 = \sum_{i,d} \frac{L}{3^i d} b_{i,d} = 24t_0, \qquad W_0 = \sum_d (M/d) b_{1,d}, \qquad T_0 = \sum_{j,d} \frac{N}{3^j d} a_{j,d}.$$

Substitution in (8.29) gives the integer identity

$$3T_0 = F_0 - 16 \cdot 3^{r-2} W_0, \qquad \Delta T_0 = 16 \cdot 3^{r-2} \sum_d (M/d) z_d. \tag{8.30}$$

It already gives $8 \mid T_0$. If $r = 2$, the kernel at $d = M$ changes T_0 by 16; choose it to make $3 \mid T_0$. If $r \geq 3$, the target infinity congruence and (8.27) imply

$$\sum_d (M/d) b_{0,d} \equiv 0 \pmod 3, \qquad W_0 \equiv 0 \pmod 3.$$

Since $d^{-1} \equiv d$ (mod 3), the first follows by reducing the infinity sum. Equation (8.30) now shows that the exact extra condition is

$$3 \mid t_0, \quad \text{equivalently } 72 \mid F_0. \tag{8.31}$$

The norm condition at the cusp 1/3 gives (8.31). On the lower curve put

$$g = \frac{\eta(3z)^{12}}{\eta(z)^{12}}, \qquad h = \frac{\eta(9z)^4}{\eta(3z)^4} \quad \text{on the upper curve.}$$

The two eta sums and the character moment show that g is a level-three unit and h a level-twenty-seven unit. They satisfy $h^3 = \beta^* g$. Since $\mathrm{ord}_\infty(g) = 1$, degree 3 proves the field identity

$$\mathbf{Q}(X_0(N)) = \mathbf{Q}(X_0(L))(\sqrt[3]{g}) \qquad (r \geq 3). \tag{8.32}$$

Write $L = 3^s M$, $s = r - 1 \geq 2$, and take the cusp 1/3, of width $w = 3^{s-2} M$ and residue field K. If q_c is its analytic width parameter, the true K-parameter is $z_c = \zeta_{3w} q_c$. This parameter follows from the Tate construction. In the chart for $\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}$, write $q_E = q_c^w$ for the elliptic Tate

period and use the cyclic subgroup represented by $z = \zeta_L q_E^{3/L}$. Its multiplicative order-three subgroup is μ_3, and the quotient marking is represented by

$$z^3 = \zeta_{L/3} q_E^{9/L} = \zeta_{3w} q_c = z_c.$$

Conversely $q_E = \zeta^{-1} z_c^w$; the quotient marking and its inverse image give the cyclic level structure over $K[[z_c]]$. Its width is w, so z_c is a uniformizer of this cusp. The eta transformation gives g analytic leading coefficient 1 and order w. Its class in the completed cusp field is

$$g = \zeta^{-1} z_c^w \pmod{\left(K((z_c))^*\right)^3}. \tag{8.33}$$

The remaining factor has a unique cube root with constant term 1 in $K[[z_c]]$. Put $B_0 = \sum_d (M/d) b_{0,d}$; it is divisible by 3. The cusp-order formula gives

$$\mathrm{ord}_{1/3}(u) = t_0 - 3^{s-3} B_0. \tag{8.34}$$

For $s \geq 3$, (8.33) is an unramified extension of the cusp valuation with residue extension $K(\zeta_9)/K$, of degree 3. The order of a norm is divisible by 3. Equation (8.34) yields (8.31).

For $s = 2$, put

$$v = \mathrm{ord}_{1/3}(u) = t_0 - B_0/3.$$

In the parameter z_c, write

$$g = g_c z_c^M (1 + O(z_c)), \qquad g_c = \zeta^{-1}, \qquad u = u_c z_c^v (1 + O(z_c)), \qquad u_c \in K^*.$$

The cusp is ramified, and its norm condition depends on these leading coefficients. Define

$$\epsilon_d = \begin{cases} 1 & d \equiv 1 \pmod 3, \\ -1 & d \equiv -1 \pmod 3, \end{cases} \qquad E_0 = \sum_d \epsilon_d b_{0,d}, \quad J = \sum_d b_{2,d}/3.$$

Here ϵ_d is defined for every integer d prime to 3. For a positive integer δ, set

$$g_\delta = \gcd(\delta, 3), \qquad A = \delta/g_\delta, \qquad C = 3/g_\delta.$$

Choose integers B, D with $AD - BC = 1$, and set

$$\gamma_\delta = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

The change of coordinate

$$w_\delta = \frac{g_\delta^2}{\delta} z - \frac{B g_\delta}{\delta}$$

satisfies

$$C w_\delta + D = \frac{g_\delta}{\delta}(3z + 1), \qquad \delta \frac{z}{3z+1} = \gamma_\delta w_\delta.$$

For $C > 0$, the eta transformation law is

$$\eta(\gamma_\delta w_\delta) = \exp\left(2\pi i \left[\frac{A+D}{24C} - \frac{s(D,C)}{2} - \frac{1}{8}\right]\right) (C w_\delta + D)^{1/2} \eta(w_\delta),$$

where

$$s(D,C) = \sum_{a=1}^{C-1} \left(\!\!\left(\frac{a}{C}\right)\!\!\right) \left(\!\!\left(\frac{Da}{C}\right)\!\!\right), \qquad ((x)) = \begin{cases} x - \lfloor x \rfloor - \frac{1}{2}, & x \notin \mathbf{Z}, \\ 0, & x \in \mathbf{Z}. \end{cases}$$

The square-root factor contributes the positive amplitude $(g_\delta/\delta)^{1/2}$. The leading exponential of $\eta(w_\delta)$ contributes $-Bg_\delta/(24\delta)$. Thus the leading phase of $\eta(\delta z)$ at the cusp, in units of $2\pi i$, is

$$\frac{A+D}{24C} - \frac{s(D,C)}{2} - \frac{1}{8} - \frac{B g_\delta}{24\delta}.$$

If $3 \nmid \delta$, take $D \in \{1, 2\}$ inverse to δ modulo 3, and $B = (\delta D - 1)/3$. The two terms $D/72$ cancel. Since

$$s(1,3) = (-1/6)^2 + (1/6)^2 = 1/18, \qquad s(2,3) = 2(-1/6)(1/6) = -1/18,$$

the remaining Dedekind term is $-\epsilon_\delta/36$. If $3 \mid \delta$, take $C = 1$, $D = 0$, and $B = -1$; the Dedekind sum is empty and equals zero. The resulting two phases are

$$\phi(3,\delta) = \begin{cases} \dfrac{\delta}{72} + \dfrac{1}{72\delta} - \dfrac{\epsilon_\delta}{36} - \dfrac{1}{8}, & 3 \nmid \delta, \\ \dfrac{\delta}{72} + \dfrac{1}{8\delta} - \dfrac{1}{8}, & 3 \mid \delta. \end{cases} \tag{8.35}$$

Summing (8.35), using weight zero, and replacing q_c by z_c, subtracts $v/(3M)$ from the phase. The reciprocal terms cancel exactly because

$$\frac{v}{3M} = \frac{1}{72}\sum_d \frac{b_{0,d}}{d} + \frac{1}{8}\sum_{\substack{i\ge 1 \\ d\mid M}} \frac{b_{i,d}}{3^i d}.$$

The remaining phase is $A_\infty/3 - E_0/36$. At a prime $\ell \neq 3$, the amplitude exponent is $-\frac{1}{2}\sum v_\ell(d) b_{i,d}$. The raw conditions make this sum divisible by 3, and the character condition makes it even, so this amplitude is a rational cube. The amplitude at 3 is $3^{-3J/2}$. Thus

$$u_c \equiv \exp\bigl(2\pi i(A_\infty/3 - E_0/36)\bigr)\, 3^{-3J/2} \pmod{\mathbf{Q}^{*3}}. \tag{8.36}$$

The unit $g/(g_c z_c^M)$ has constant term 1, so its binomial series gives a cube root in $K[[z_c]]$. After dividing the Kummer generator by that root, the local extension has equation

$$t^3 = g_c z_c^M.$$

Choose integers A, B with $AM + 3B = 1$, and put

$$s_c = t^A z_c^B.$$

Then

$$s_c^3 = g_c^A z_c, \qquad \mathrm{Nm}(s_c) = (g_c z_c^M)^A z_c^{3B} = g_c^A z_c.$$

The polynomial $X^3 - g_c^A z_c$ is Eisenstein over $K[[z_c]]$. Thus the extension is totally ramified of degree 3, with uniformizer s_c and residue field K. The norm preserves normalized valuations, since the residue degree is one. If an upper function has norm u, its leading term is $b s_c^v$, with $b \in K^*$; the norm of its remaining unit has constant term 1. Hence

$$u_c = g_c^{Av} b^3, \qquad \frac{g_c^v}{u_c^M} = g_c^{v(1-AM)} b^{-3M} = \bigl(g_c^{Bv} b^{-M}\bigr)^3.$$

In particular g_c^v/u_c^M is a cube in K. With $\sqrt{-3} = \zeta - \zeta^2$, one has $(\sqrt{-3})^{3MJ} = \zeta_{36}^{27MJ} 3^{3MJ/2}$. Equation (8.36) therefore gives

$$\frac{g_c^v}{u_c^M} \equiv (\sqrt{-3})^{3MJ} \zeta_{36}^{A_0} \pmod{\mathbf{Q}^{*3}},$$

where

$$A_0 = -12v - 12MA_\infty + ME_0 - 27MJ. \tag{8.37}$$

The root $\zeta_{36}^{A_0}$ lies in K, so $6 \mid A_0$; it is a cube in K exactly when $18 \mid A_0$, since the roots of unity in K are μ_6. Put

$$R_0 = A_0 + 12t_0 = 4B_0 - 12MA_\infty + ME_0 - 27MJ.$$

Reducing in $\mathbf{Z}_{(3)}$, using $b_{1,d} \equiv -b_{0,d} \pmod 3$, gives

$$R_0/M \equiv \sum_d (4d^{-1} + d + \epsilon_d) b_{0,d} \pmod 9.$$

The residues are

$d \bmod 9$	1	2	4	5	7	8
$d^{-1} \bmod 9$	1	5	7	2	4	8
ϵ_d	1	−1	1	−1	1	−1
$4d^{-1} + d + \epsilon_d \bmod 9$	6	3	6	3	6	3

Thus $R_0/M \equiv 6E_0 \pmod 9$. The infinity sum gives $E_0 \equiv 0 \pmod 3$, so $9 \mid R_0$. Also $6 \mid R_0$, hence $18 \mid R_0$. The norm condition $18 \mid A_0$ now implies $3 \mid t_0$. This proves (8.31) when $s = 2$.

We can therefore choose the integral source vector (8.29), with its kernel correction, to satisfy every condition (8.24). The corresponding source unit has norm u: the Euler exponents, infinity order, and leading coefficient all agree. Restoring the initial constant norm proves (8.22). □

8.5. **Cyclotomic unit norms.**

Theorem 8.11. *For $N = 3^r M$, $r \geq 2$, $3 \nmid M$, and $L = N/3$,*

$$\mathcal{U}_L(K) \cap \mathrm{Nm}_\beta K(X_0(N))^* = \mathrm{Nm}_\beta \mathcal{U}_N(K). \tag{8.38}$$

This equality includes constants.

Proof. For $L = 3^s M$, every leading-normalized K-rational cusp unit has a unique expansion

$$u = q^A \prod_{\substack{e \mid M \\ 0 \leq i \leq s}} E(q^{3^i e})^{b_{i,e}} \prod_{\substack{e \mid M \\ 0 \leq j \leq s-2}} E(\zeta q^{3^j e})^{d_{j,e}}, \qquad b_{i,e}, d_{j,e} \in \mathbf{Z}, \tag{8.39}$$

where the second product is empty if $s = 1$. Its eta weight is zero and

$$\sum b_{i,e} + \sum d_{j,e} = 0, \qquad 24A = \sum 3^i e\, b_{i,e} + \sum 3^j e\, d_{j,e}. \tag{8.40}$$

These exponent vectors satisfy the modular multiplier conditions.

At a cusp a/c, $(a, c) = 1$, of width w_c, the exact eta orders are

$$\begin{aligned} \mathrm{ord}_{a/c}\, \eta(mz) &= \frac{w_c}{24m} \gcd(m, c)^2, \\ \mathrm{ord}_{a/c}\, \eta(mz + k/3) &= \frac{w_c}{216m} \gcd(3ma + kc, 3c)^2 \qquad (k = 1, 2). \end{aligned} \tag{8.41}$$

For the second equality, the transformed integral matrix has determinant $9m$; factor the gcd of its first column and apply the eta transformation law. This also proves the formula with arbitrary tame powers in c and m.

Over K, a rational cusp orbit splits into two exactly at the interior three-heights $1, \ldots, s-1$. For

$$R_m(q) = \frac{E(\zeta q^m)}{E(\zeta^2 q^m)}, \qquad m = 3^j e, \quad 0 \leq j \leq s-2, \quad e \mid M,$$

the two orders in (8.41) differ only at height $i = j + 1$. At that height, with $c = 3^{j+1} d$, their first-column gcds are $3^{j+1} \gcd(e, d)$ and $3^{j+2} \gcd(e, d)$, in one order. The difference of cusp orders is

$$\pm w_c\, 3^{j-1} \frac{\gcd(e, d)^2}{e}.$$

The sign factors as a row sign and a column sign, determined by the residues of a, d, e modulo 3. Thus the anti-divisor matrix is block diagonal by height, and each block is an invertible row and column rescaling of the matrix $\left(\gcd(e, d)^2\right)_{e,d \mid M}$. Its invertibility for every M follows from the exact incidence factorization

$$\gcd(e, d)^2 = \sum_{k \mid e,\ k \mid d} J_2(k), \qquad J_2(k) = k^2 \prod_{\ell \mid k} (1 - \ell^{-2}) > 0. \tag{8.42}$$

The incidence matrix of divisibility is triangular with diagonal entries 1. The ordinary eta matrix is an invertible row and column rescaling of the same gcd-square matrix on all divisors of L. Consequently ordinary etas and the shifts span all K-rational cusp divisors over $\mathbf{Q}$; degree zero gives weight zero.

Conjugation by $z \mapsto mz + 1/3$ gives the matrix (8.55), with $N = L$. It is integral when $9m \mid L$, since the diagonal entries of a matrix in $\Gamma_0(L)$ agree modulo 3. Its eta multiplier has finite order. Clear the rational exponents and these multipliers. Comparison of divisors on the proper curve gives an equality $u^a = F$ after leading normalization, where F is an integral product of ordinary

and shifted Euler factors. The normalized u has bounded denominators: multiplying by a power of Δ gives a holomorphic congruence form with coefficients in a number field. At any finite place of K, the Gauss valuation is multiplicative on such bounded series. Since F has constant term 1 and integral coefficients, it has Gauss valuation zero, and so does the normalized u. Its coefficients are integral at every place.

Write $b_m = b_{i,e}$ for $m = 3^i e$, and $d_m = d_{j,e}$ for $m = 3^j e$, taking missing entries to be zero. Order these indices m increasingly. After removing the factors already determined, the first coefficient at q^m is $-b_m - \zeta d_m$, with the original exponents rational. The equality $\mathcal{O}_K = \mathbf{Z} + \mathbf{Z}\zeta$ forces both coefficients to be integers. Induction proves (8.39) and its uniqueness.

If u is a function norm, its infinity leading coefficient is a^3, $a \in K^*$, because $q = t^3$ at the rational fixed cusp and $\operatorname{Nm} t = q$. Divide by the norm of this constant. Henceforth every expression has leading coefficient 1.

Put

$$\lambda = \zeta - 1, \qquad R = \widehat{(\mathcal{O}_{K,\lambda}[[q]])}_{(\lambda)}, \qquad F = \operatorname{Frac}(R).$$

As in the rational proof, the cover becomes the extension $t^3 = q$, with integral closure $S = R[t]$. Its residue extension is purely inseparable of degree 3, and $\operatorname{Tr}(S) \subset 3R$. For $z \in S$, the second elementary symmetric function satisfies

$$2e_2(z) = \operatorname{Tr}(z)^2 - \operatorname{Tr}(z^2),$$

so $e_2(z) \in 3R$. Therefore

$$\operatorname{Nm}(1 + \lambda z) = 1 + \lambda \operatorname{Tr} z + \lambda^2 e_2(z) + \lambda^3 \operatorname{Nm} z \equiv 1 \pmod{\lambda^3}. \tag{8.43}$$

If a norm is $1 \pmod{\lambda}$, its witness has valuation zero and residue whose cube is 1; the residue is therefore 1, and (8.43) applies.

Form the local expression

$$V(t) = t^A \prod_m E(t^m)^{b_m + d_m} \in F(t).$$

Its norm agrees with u modulo λ. Moreover, the rational series

$$\frac{\operatorname{Nm} V}{q^A \prod_m E(q^m)^{b_m + d_m}} \equiv 1 \pmod 3$$

is also $1 \pmod{\lambda^2}$. On the other hand,

$$\frac{E(\zeta q^m)}{E(q^m)} \equiv 1 - \lambda \sum_{k \geq 1} \frac{k q^{mk}}{1 - q^{mk}} \pmod{\lambda^2}. \tag{8.44}$$

The series on the right starts with $-q^m$. The least-index argument, now applied to the coefficient of λ, gives

$$d_{j,e} \equiv 0 \pmod 3 \quad \text{for every } j, e. \tag{8.45}$$

The local identity

$$\operatorname{Nm} E(\zeta t^{3m}) = E(\zeta q^m)^3$$

removes all shifted factors. The remaining rational Euler product is still a local norm. Remove its rational canonical norm, as in the preceding proof. The quotient is $1 \pmod 3$; (8.43) makes it $1 \pmod{\lambda^3}$. Its coefficients are rational, whose λ-valuations are even, so it is actually $1 \pmod 9$. The calculation with $E(q^m)^3 / E(q^{3m})$ in the rational proof gives

$$b_{0,e} + b_{1,e} \equiv 0 \pmod 3, \qquad b_{i,e} \equiv 0 \pmod 3 \quad (i \geq 2). \tag{8.46}$$

For a norm at the shallow level, put on $X_0(9)$

$$w = \left(\frac{\eta(z)}{\eta(9z)} \right)^3, \qquad a = 3(\zeta - 1), \quad b = 3(\zeta^2 - 1), \quad \rho_0 = \zeta - \zeta^2.$$

The coordinate w has one simple pole at ∞ and one simple zero at the cusp 0. The level-three function g in (8.32) has its only pole, of order 3, at $w = 0$, and its remaining cusp orders give a quadratic numerator. Its normalized expansion gives the exact identity

$$g = \frac{w^2 + 9w + 27}{w^3} = \frac{(w-a)(w-b)}{w^3}. \tag{8.47}$$

Indeed the Euler products give

$$w = q^{-1} - 3 + 5q^2 + O(q^5), \qquad g = q + 12q^2 + 90q^3 + O(q^4),$$

and hence $gw^3 = q^{-2} + 3q^{-1} + 9 + O(q)$. If its numerator is $Aw^2 + Bw + C$, comparison gives $A = 1$, $-6A+B = 3$, and $9A-3B+C = 9$. Thus $B = 9$, $C = 27$; the pole and degree constraints exclude further terms.

On $X_0(27)$, put $W = \beta^* w$, $x_{27} = Wh$, and $Y = 2W + 9$, with h as in (8.32). Then

$$x_{27}^3 = (W-a)(W-b), \qquad Y^2 = 4x_{27}^3 - 27. \tag{8.48}$$

Its six cusps are exactly

$$\infty, \quad P_a = (0, 3\rho_0), \quad P_b = (0, -3\rho_0), \quad Q_k = (3\zeta^k, 9) \quad (k = 0, 1, 2).$$

Indeed the three cusps $W = \infty, a, b$ are ramified, and $W = 0$ has precisely the three displayed preimages. Define

$$f = \frac{2(x_{27} - 3)}{Y - 3\rho_0 - 2(\zeta^2 - 1)x_{27}}. \tag{8.49}$$

The vertical line $x_{27} = 3$ meets the cubic at $Q_0, -Q_0$; the denominator line meets it at $P_a, Q_1, -Q_0$. All three intersections of the latter line are simple, as follows by substitution in (8.48). The common point $-Q_0$ cancels, and the pole orders of a line and a vertical line at infinity give

$$\mathrm{Div}(f) = Q_0 + \infty - Q_1 - P_a.$$

Thus f is a cusp unit. Multiplying over the three translates $x_{27}, \zeta x_{27}, \zeta^2 x_{27}$ gives

$$\begin{aligned} \mathrm{Nm}(x_{27} - 3) &= x_{27}^3 - 27 = W(W+9), \\ \mathrm{Nm}(Y - 3\rho_0 - 2(\zeta^2-1)x_{27}) &= (Y - 3\rho_0)^3 - 8(\zeta^2-1)^3 x_{27}^3 \\ &= 8(W-a)W(W+9), \\ \mathrm{Nm}(f) &= (W-a)^{-1}. \end{aligned} \tag{8.50}$$

In the first two equations use $x_{27}^3 = W^2 + 9W + 27$ and $Y = 2W + 9$; these polynomial identities hold modulo $\zeta^2 + \zeta + 1$. Conjugation and division show that

$$U = \frac{w-a}{w-b}$$

is a cusp-unit norm.

The orders from (8.41) show that R_1^3 has order 0 at $0, \infty$ and orders $-1, +1$ at $1/3, 2/3$, respectively. Its eta multiplier is finite. The group $\Gamma_0(9)/\{\pm 1\}$ is torsion free of genus zero and is generated by cusp parabolics; these integral orders make that multiplier trivial. The eta phase gives $w(1/3) = b$, $w(2/3) = a$. Thus the divisors of U and R_1^3 agree; their leading coefficient is 1. The coefficient of q is $-3(\zeta - \zeta^2)$ on both sides, also fixing the orientation. Hence

$$U = R_1^3.$$

For each $e \mid M$, pull back by $z \mapsto ez$. This gives a degree-three base change because $\mathrm{ord}_\infty g(ez) = e$ is prime to 3; its cube root lies on the upper modular curve. Therefore $U_e = R_e^3$ is a cusp-unit norm at every tame axis. The Euler identity

$$E(q^e)E(\zeta q^e)E(\zeta^2 q^e) = E(q^{3e})^4 / E(q^{9e})$$

gives its mixed exponent vector

$$U_e = E(q^e)^3 E(q^{3e})^{-12} E(q^{9e})^3 E(\zeta q^e)^6. \tag{8.51}$$

Its shifted exponent in this basis is 6.

Now let $r = 3$, so the lower level is $9M$. Write $d_e = d_{0,e}$. By (8.45), $d_e/3$ is integral. Set

$$v = u^2 \prod_{e|M} U_e^{-d_e/3}. \tag{8.52}$$

Every factor is a K-unit and a K-function norm. The shifted exponent is $2d_e - 6d_e/3 = 0$. The remaining ordinary exponents are exactly

$$(\, 2b_{0,e} - d_e, \quad 2b_{1,e} + 4d_e, \quad 2b_{2,e} - d_e \,).$$

Thus v has rational expansion with leading coefficient 1, and the q-expansion principle makes it a rational cusp unit. Choose $f_0 \in K(X_0(N))^*$ with $\mathrm{Nm}_\beta f_0 = v$. The rational function $f_0\tau(f_0)$ has rational norm v^2. Proposition 8.10 gives $H \in \mathcal{U}_N(\mathbf{Q})$ with $\mathrm{Nm}\, H = v^2$. Consequently

$$\mathrm{Nm}\, \frac{H^2}{\beta^* v} = v^{4-3} = v.$$

Combining this witness with the witnesses for (8.51) in (8.52) gives $G \in \mathcal{U}_N(K)$ with $\mathrm{Nm}\, G = u^2$. Finally

$$\mathrm{Nm}\, \frac{G^2}{\beta^* u} = u^{4-3} = u. \tag{8.53}$$

This proves the case $r = 3$. For $r = 2$ there are no shifted indices. The normalized unit is already rational, and the same constant-field norm and $4 - 3$ argument proves the result.

It remains to treat $s = r - 1 \geq 3$. The multiplier of η^8 is the cubic character χ of $\mathrm{SL}_2(\mathbf{Z})$ with $\chi(T_0) = \zeta$, $\chi(S_0) = 1$. It factors through $\mathrm{SL}_2(\mathbf{F}_3)$. For a matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ over $\mathbf{F}_3$, its exponent is

$$\chi_{\exp}(a, b, c, d) = \begin{cases} ab, & c = 0, \\ (a + d)/c, & c \neq 0. \end{cases} \tag{8.54}$$

Indeed, when $c = 0$, the matrix is $\mathrm{diag}(a, a^{-1})T_0^{b/a}$; the diagonal matrix is $\pm I$. When $c \neq 0$, its Bruhat decomposition is

$$T_0^{a/c}\, \mathrm{diag}(c^{-1}, c) S_0 T_0^{d/c}.$$

This proves (8.54) for all 24 matrices of $\mathrm{SL}_2(\mathbf{F}_3)$.

Let $\gamma = \begin{pmatrix} a & b \\ Nu & d \end{pmatrix} \in \Gamma_0(N)$ and $k = (d - a)/3$. Conjugation by $z \mapsto mz + 1/3$ gives the integral determinant-one matrix

$$\begin{pmatrix} a + Nu/(3m) & mb + (d - a)/3 - Nu/(9m) \\ Nu/m & d - Nu/(3m) \end{pmatrix}. \tag{8.55}$$

Its automorphy factor for η^8 is exactly $(Nuz + d)^4$. For $m = 3^j e$, $j \leq r - 2$, equation (8.54) gives exponent

$$a[3^j e\, b + k - 3^{r-j-2}(M/e)u].$$

The ordinary factors contribute aeb at height 0, $-a(M/e)u$ at height r, and 0 at the other heights. Thus a weight-zero source product with ordinary exponents $a_{j,e}$ and shifted exponents $c_{j,e}$ has an eighth power with total multiplier

$$a(bA_0 - uB_0 + kC_0) \pmod 3,$$

where

$$A_0 = \sum_e e(a_{0,e} + c_{0,e}), \quad B_0 = \sum_e (M/e)(a_{r,e} + c_{r-2,e}), \quad C_0 = \sum_{j,e} c_{j,e}. \tag{8.56}$$

It suffices to make all three sums zero modulo 3.

The ordinary and shifted norm stencil is obtained from (8.26) and

$$\operatorname{Nm} E(\zeta t^m) = \begin{cases} E(q^m)^4/E(q^{3m}), & 3 \nmid m, \\ E(\zeta q^{m/3})^3, & 3 \mid m. \end{cases}$$

For the target vector in (8.39), an integral preimage is

$$\begin{aligned} x_e := a_{0,e} + c_{0,e} &= -b_{1,e} + 3z_e, \\ a_{1,e} &= (b_{0,e} - 4x_e)/3, \\ a_{2,e} &= (b_{1,e} + x_e)/3, \\ a_{i+1,e} &= b_{i,e}/3 \quad (i \geq 2), \\ c_{j+1,e} &= d_{j,e}/3, \end{aligned} \tag{8.57}$$

with the $c_{0,e}$ free. The raw congruences make every entry integral. The source weight is one third of the target weight, hence zero; its infinity order equals the target integer A. The infinity condition and the raw congruences give $A_0 = 0 \pmod 3$. Choose $c_{0,M}$ to make $C_0 = 0 \pmod 3$, adjusting $a_{0,M} = x_M - c_{0,M}$. The sole remaining condition is

$$\sum_{e \mid M} (M/e)(b_{s,e} + d_{s-2,e}) \equiv 0 \pmod 9. \tag{8.58}$$

Let v_0, v_3 be the orders of u at the lower cusps $0, 1/3$. Formula (8.41) gives

$$v_3 - v_0 = 3^{s-3} \left[-\sum_e (M/e) b_{0,e} + \frac{1}{2} \sum_e (M/e)(1 - \epsilon_e) d_{0,e} \right]. \tag{8.59}$$

For the ordinary factors only height 0 changes. For a shifted factor, the gcd of the first column of $\begin{pmatrix} 3m+3 & 1 \\ 9 & 3 \end{pmatrix}$ gives the same order as at 0 when $j \geq 1$; at $j = 0$ it also agrees if $\epsilon_e = 1$, and increases by $3^{s-3} M/e$ if $\epsilon_e = -1$. This proves (8.59) term by term. Its bracket is divisible by 3, by (8.45), (8.46), and the infinity condition. The cusp Kummer equation (8.33) has residue extension $K(\zeta_9)/K$ at $s \geq 3$. The function-norm hypothesis forces $3 \mid v_3$. Hence (8.59) gives $3 \mid v_0$.

Finally the exact reciprocal order is

$$24 v_0 = \sum_{i,e} \frac{L}{3^i e} b_{i,e} + \sum_{j,e} \frac{L}{9 \cdot 3^j e} d_{j,e}. \tag{8.60}$$

Modulo 9, every term except the two top terms in (8.58) vanishes. Since $3 \mid v_0$, that condition follows. All three multipliers in (8.56) are now zero.

Let F_η be the leading-normalized source eta expression supplied by (8.57). We have proved that F_η^8 is a K-cusp unit. Its norm is u^8, since the Euler exponents, infinity order, and leading coefficient agree. The unit

$$\frac{F_\eta^{16}}{(\beta^* u)^5} \tag{8.61}$$

has norm $u^{16-15} = u$. Restoring the initial constant norm proves (8.38) at every depth. □

8.6. **The cyclic quotients and integral model.** Return to $N = 3^n M$, $n \geq 3$, $3 \nmid M$. Over K, the adjacent maps

$$\alpha : X_0(N) \to X_0(N/3), \quad z \mapsto z, \qquad \beta : X_0(N) \to X_0(N/3), \quad z \mapsto 3z$$

are cyclic of degree 3. We choose their generators consistently with the matrices

$$S = \begin{pmatrix} 1 & 0 \\ -N/3 & 1 \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 1/3 \\ 0 & 1 \end{pmatrix}. \tag{8.62}$$

The induced actions on functions and divisor classes are chosen so that taking a divisor commutes with the action. Replacing a generator by its inverse changes none of the assertions below. The generators commute, have order 3, and satisfy

$$\tau S\tau = S^{-1}, \qquad \tau T\tau = T^{-1}.$$

Both adjacent lower curves have level divisible by 9. They have no elliptic stabilizer: $d^2+1=0$ has no solution modulo 3, and $d^2+d+1=0$ has none modulo 9. Thus both coordinate covers are unramified away from cusps. At their rational fixed cusps,

$$\alpha^*(0) = 3(0), \qquad \beta^*(\infty) = 3(\infty), \tag{8.63}$$

with no residue extension. Put

$$D = (1-S)(1-T) = (S-1)(T-1).$$

We will prove $DQ = 0$ for every $Q \in J_0(N)(K)[3^\infty]$.

Proposition 8.12. *Let $Q \in J_0(N)(K)[3^\infty]$. On an auxiliary full-four fine curve at 3, the pullback of DQ, and every S^iT^jDQ, extends to the base change of the original unramified Picard identity. All these extensions have the same special class. That class lies in*

$$\operatorname{im}\bigl(H^1(F, 1+\mathcal{J}) \to \operatorname{Pic}(F)\bigr), \qquad \mathcal{J}^2 = 0,$$

where F is the entire original special fiber and $\mathcal{J}$ is the ideal supported on the two shallow components described below.

Proof. Choose a sufficiently large finite unramified coefficient DVR $R_0/\mathbf{Z}_3$, containing the required prime-to-three marking roots and the full-four pairing, and put $k = R_0/3R_0$. Let $\mathscr{X}_0/R_0$ be the regular compact fine model corresponding to $\Gamma_0(N)\cap\Gamma(4)$, on its chosen connected component. The cyclic moduli problem with this representable étale auxiliary marking is regular of dimension two by [25, 6.6.1–6.6.2]; its compactification is the normalization finite over the original j-line [25, 8.6.2–8.6.3]. The Tate charts are regular at the cusps.

A cyclic chain has the form

$$E_0 \longrightarrow E_1 \longrightarrow \cdots \longrightarrow E_n.$$

Forget its first and last arrows, retaining $E_1 \to \cdots \to E_{n-1}$, and transport the tame marking and full-four basis along the first isogeny. Correct the resulting cubed pairing by the fixed inverse of 3 modulo 4. This gives a map to the fine curve of depth $n-2$. It is finite. Off the cusps, recovering the two omitted arrows is a finite representable three-isogeny problem [25, 6.5.1]; imposing that their composite is cyclic and has the prescribed standard factors is a closed condition in the product of these finite problems [25, §6.7]. At a cusp the Tate parameters satisfy

$$q_{\text{lower}} = \xi q_{\text{upper}}^{e}, \qquad e > 0,$$

over a finite coefficient extension. The completed cusp rings are finite normalizations of these Tate rings. Thus the finite map extends to the same compactifications. Equivalently, each special component maps by a nonconstant Frobenius power of the middle ordinary modulus, and the finite cusp charts exclude any remaining positive-dimensional fiber.

The compatible fine lifts are

$$S_4 = \begin{pmatrix} 1 & 0 \\ -4N/3 & 1 \end{pmatrix}, \qquad T_4 = \begin{pmatrix} 1 & 4/3 \\ 0 & 1 \end{pmatrix}.$$

With $A = \operatorname{diag}(3,1)$,

$$AS_4A^{-1} = \begin{pmatrix} 1 & 0 \\ -4N/9 & 1 \end{pmatrix}, \qquad AT_4A^{-1} = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}. \tag{8.64}$$

Both matrices lie in the depth-$n-2$ fine group. Their cubes lie in the upper fine group. For $N = 27m$, their commutator is exactly

$$\begin{pmatrix} 1+48m & -64m \\ -1728m^2 & 1-48m+2304m^2 \end{pmatrix}, \tag{8.65}$$

which also lies in that group. The index of the upper in the common lower group is 9. Hence the two commuting order-three actions give precisely its generic quotient. Formula (8.65) includes $n = 3$. For T_4, the full-four cusp parameter changes by $q_4 \mapsto \zeta q_4$. For S_4, the full-level Galois rule conjugates its integral lower unipotent matrix by a diagonal cyclotomic matrix; modulo the descending group only the exponent modulo 3 matters. These statements give definition over K and inversion by τ for both fine actions, including even M.

Put $R = R_0[\zeta]$, let $\mathscr{X} = \mathscr{X}_0 \otimes R$, and let $\mathscr{Y}$ be its finite normalization. The preceding finite common lower map identifies $\mathscr{Y}$ with the integral closure of that lower model in the upper generic field. Both generic deck transformations extend to $\mathscr{Y}$, over the common lower model. We will use their actions there.

At a supersingular point, the fine cyclic local equations [25, 13.4.7] give two outer branches with initial linear terms x, y; they form a regular parameter pair in the original regular two-dimensional ring A_0. The multiplicities of the inner types are $2 \cdot 3^{\min(a,n-a)-1}$, while the two outer multiplicities are 1. Unique factorization in A_0 therefore gives

$$-3 = u\, xyG^2.$$

Absorb the unit u into x. For $\lambda' = \sqrt{-3} = \zeta - \zeta^2$, the original ramified ring and its normalization are

$$A = A_0[\lambda']/(\lambda'^2 - xyG^2), \qquad B = A_0[w]/(w^2 - xy), \qquad \lambda' = Gw. \tag{8.66}$$

The ring B is normal. The finite flat ring

$$C = A_0[a, b]/(a^2 - x, b^2 - y)$$

has local dimension two and maximal ideal generated by a, b, hence is regular at the relevant points. The simultaneous sign involution $(a, b) \mapsto (-a, -b)$ has invariant ring $A_0[ab] = B$; its order is a unit. Thus B is normal and, being finite and generically isomorphic to A, is its normalization. Pulling a Weil divisor to the factorial local ring C and taking the norm of a principalization proves $2\,\mathrm{Cl}(B) = 0$. Where only one outer branch occurs, the normalization is a regular square-root parameter cover; where neither occurs, it is étale locally regular. The normalized compact cusp rings are regular cyclotomic coefficient DVRs with one formal parameter. All local Cartier obstructions on $\mathscr{Y}$ are therefore killed by 2.

The generic odd-order bundle representing the fine pullback of Q extends to a line bundle on $\mathscr{Y}$. Indeed, take the closure E of a generic divisor and let $m = 3^a$ kill its generic class. For some rational function f_0,

$$mE - \mathrm{Div}(f_0) = V_0$$

is vertical. In every local class group, $[E - V_0] = (1-m)[E] = 0$, because m is odd and the class group is killed by 2. Thus $E - V_0$ is Cartier.

The conductor inclusions needed below follow directly from (8.66):

$$\lambda' B \subset A, \qquad (\lambda')^2 B \subset \lambda' A. \tag{8.67}$$

In fact $\lambda'(a_0 + b_0 w) = \lambda' a_0 + Gxyb_0$. The uniformizers $\lambda' = -\zeta\lambda$ and $\lambda = \zeta - 1$ differ by a unit, so the same ideal inclusions hold with λ.

For a mixed ordinary type (a, b), let $a, b \geq 1$ and $a + b = n$. Write x for the source Serre–Tate period. In the standard coordinates for its extension of $\mathbf{Q}_3/\mathbf{Z}_3$ by μ_{3^∞}, the subgroup has connected part μ_{3^a} and étale quotient of order 3^b. Locally choose a generator with coordinates

$(U, 1/3^b)$, and put $y = U^{3^a}$. This is independent of the choice of U modulo μ_{3^a}. The cyclicity equation is

$$\Phi_3\Big(y^{3^{b-1}}/x^{3^{a-1}}\Big) = 0, \qquad \Phi_3(U) = U^2 + U + 1 \tag{8.68}$$

in the ordinary marked coordinates of [25, 13.6.5–13.6.6]. It gives $y^{3^b} = x^{3^a}$. Quotienting by μ_{3^a} changes the standard period relation $(x, 1)$ to $(x^{3^a}, 1)$. The remaining quotient adds the relation $(y, 1/3^b)$. Multiplying the second coordinate by 3^b gives the relation $(y, 1)$, so the target period is y. For $b \geq 2$, forgetting the last arrow on the generic fiber replaces a cyclic generator by its triple. Taking the finite flat closure gives

$$\alpha : (x, y) \longmapsto (x, y^3), \qquad S : (x, y) \longmapsto (x, \zeta y).$$

For $a \geq 2$, forgetting the first arrow quotients by μ_3, so

$$\beta : (x, y) \longmapsto (x^3, y), \qquad T : (x, y) \longmapsto (\zeta x, y).$$

These formulas describe the operations whose lower chart is mixed; the outer cases are treated below. If $a \leq b$, its normalized phase and parameter can be written

$$\xi = y^{3^{b-a}}/x, \qquad t = y, \qquad \xi \text{ primitive of order } 3^a.$$

For $d = b - a \geq 0$, the nonexceptional source operation and target operation are, respectively,

	source	target
$d > 0$	$(\xi, t) \mapsto (\zeta^{-1}\xi, t)$	$(\xi, t) \mapsto (\xi, \zeta t)$
$d = 0$	$(\xi, t) \mapsto (\zeta^{-1}\xi, t)$	$(\xi, t) \mapsto (\zeta\xi, \zeta t)$.

If $a > b$, put instead $\xi = y/x^{3^{a-b}}$, $t = x$. Then the source changes (ξ, t) to $(\xi, \zeta t)$ and the target changes it to $(\zeta\xi, t)$. When $a, b \geq 2$, every displayed operation is the identity modulo λ on the full normalized ring, including its nilpotent quotient modulo λ. The reduction uses the fixed base-ring uniformizer λ and retains the primitive $3^{\min(a,b)}$-phase.

At $a = 1, b = n - 1$, the two shallow sheets and the outer sheet are respectively

$$x = \zeta^{-i} t^{3^{n-2}} \quad (i = 1, 2), \qquad x = t^{3^n}. \tag{8.69}$$

The common middle period is $t^{3^{n-1}}$ in all three cases: for the outer sheet one deletes an étale end, and for the shallow sheets one deletes a multiplicative end. The exceptional deck therefore cycles these three sheets with t fixed. The opposite deck acts on each by the same $t \mapsto \zeta t$. Since $n - 2 \geq 1$, this preserves the displayed source periods. It also agrees with commutation of the two decks. In particular the outer sheet in (8.69) is distinct from a shallow formula with phase 1. At the other end, $a = n - 1, b = 1$, put $t = x$. The two shallow sheets and the outer sheet are

$$y = \zeta^i t^{3^{n-2}} \quad (i = 1, 2), \qquad y = t^{3^n}.$$

Their common lower target period is $t^{3^{n-1}}$. Here S cycles the three sheets with t fixed, and T acts on each by $t \mapsto \zeta t$.

Let C_0 denote the normalized coefficient algebra at this end. The normalized three-sheet algebra is C_0^3, and the original order is

$$C_0 \ \oplus \ \{(f_+, f_-) \in C_0^2 : f_+ \equiv f_- \pmod{\lambda}\}. \tag{8.70}$$

Evaluation of $C_0[\omega]/(\omega^2 + \omega + 1)$ at $\omega = \zeta, \zeta^2$ proves this description.

For a unit g on the normalized inverse image of a common lower open, form the exact product

$$D(g) = \frac{g\, S(Tg)}{(Sg)(Tg)}. \tag{8.71}$$

At a nonexceptional ordinary primary, write $Tg/g = 1 + \lambda b$. Commutation and the fact that S is the identity modulo λ give $D(g) \in 1 + \lambda^2 B$. By (8.67) this lies in $1 + \lambda A$. The same holds for its inverse and every deck translate.

At a three-sheet end, write the good deck as $\sigma = 1 + \lambda\partial \pmod{\lambda^2}$, and put

$$a_i = \frac{\partial \bar{g}_i}{\bar{g}_i}, \qquad i = 0, +, -.$$

For any normalized unit, the first coefficients of (8.71) are

$$(v_0, v_+, v_-) = (a_+ - a_0,\ a_- - a_+,\ a_0 - a_-). \tag{8.72}$$

They sum to zero. Every coordinate of $D(g)$ is $1 \pmod{\lambda}$, so it and its inverse belong to the original order (8.70). To compute its original special value, interpolate the shallow pair as $A + B\omega$. Since $\zeta - \zeta^2 = -\zeta\lambda$, the coefficient of $\epsilon = \omega - 1$ modulo λ is $v_- - v_+$. The coefficients for the three cyclic translates are

$$a_0 + a_+ - 2a_-, \qquad -2a_0 + a_+ + a_-, \qquad a_0 - 2a_+ + a_-. \tag{8.73}$$

They are equal in characteristic 3. The good deck is the identity on this residue algebra. Thus all nine $S^i T^j D(g)$ lie in the original order and have the same special value. That value is $1 + \epsilon c$, with c given by (8.73). At the other end interchange S, T.

These computations extend through every complete supersingular and cusp ring. The ramified original base change is a hypersurface over the regular model, hence is S_2. At every horizontal height-one point, the generic deck action makes (8.71) and its inverse regular. At every vertical height-one point the preceding full ordinary-ring calculation does the same. The intersection description of an S_2 ring inside its total quotient ring therefore puts all these products and inverses in the original ring.

Let F be the original special divisor, and let C_L, C_R be the two reduced shallow prime components, each occurring in F with multiplicity 2. Set

$$H = F - C_L - C_R, \qquad \mathcal{J} = I_H / I_F. \tag{8.74}$$

Here I_H and I_F are the ideal sheaves of the effective Cartier divisors H and F. Locally, if $F = G_0 t_L^2 t_R^2$, then $\mathcal{J}$ is generated by $G_0 t_L t_R$ modulo F; in particular $\mathcal{J}^2 = 0$. The special products above are 1 on the outer and deeper primaries and differ from 1 by the indicated first nilpotent on the two shallow primaries. Thus $D(g) - 1 \in \mathcal{J}$ at every primary generic point. Both $\mathcal{O}_F$ and $\mathcal{O}_F/\mathcal{J} = \mathcal{O}_H$ are effective Cartier-divisor rings in a regular surface, hence Cohen–Macaulay with no embedded points. Equality of the translated special values, and membership in $\mathcal{J}$, consequently extend over the whole nonreduced fiber.

Finally choose frames for the extended normalized line bundle. They can be chosen on inverse images of common lower opens: at a lower stalk the finite inverse algebra is semilocal, every invertible module is free, and a generator extends after shrinking. Apply D to all its transition units. The operation is multiplicative, so it preserves cocycle and change-of-frame identities. The proved integrality gives a line bundle on $\mathscr{X}$ whose generic class is DQ. The same holds for all deck translates. Their special class lies in $H^1(F, 1 + \mathcal{J})$; since $1 + \mathcal{J}$ is the additive coherent sheaf $\mathcal{J}$, its image is connected and lies in $\mathrm{Pic}^0(F)$. The generic degree is zero.

The original curve $\mathscr{X}_0$ is proper, flat, and regular with connected geometric generic fiber and a smooth multiplicity-one cusp. Its special multiplicities therefore have gcd 1. The original Picard comparison of [39, 8.1.4, 8.2.1], as recalled in the foundational discussion, identifies its smooth separated Pic^0 with the original Néron identity. Its base change to R is the group in which the constructed sections lie. All translated sections have the same special class. □

8.7. **Formal subgroups and fixed branches.**

Lemma 8.13. *Let $w \in J_0(N)(K)[3]$, $n \geq 3$, satisfy $\tau w = -w$. Suppose its injective full-four pullback belongs to the formal kernel of the base change of the original unramified Picard identity in Proposition 8.12. Then $w \in \Sigma(N)[3](K)$, and its diamond character has conductor prime to 3.*

Proof. If $w = 0$ there is nothing to prove. Otherwise its cyclic subgroup is a $\mathbf{Q}$-subgroup of type μ_3. The fine pullback is injective on geometric 3-primary points: for odd N the full-four deck group is S_4 with abelianization C_2, and for even N it is a 2-group. The character-kernel argument in Proposition 8.5 applies.

We construct its finite flat closure in the original smooth group over R_0. Choose an affine neighborhood of its identity. All three points after base change to R are integral and reduce to that identity, so their sections lie entirely in the affine neighborhood. Map its coordinate algebra to the generic coordinate algebra of μ_3. Every image is integral over R_0, because all three evaluations are integral after adjoining ζ. The integral closure of R_0 in that finite algebra is a finite module. Its coordinate image is a submodule and is finite and torsion free over the Noetherian DVR. Its generic rank is 3. The group law, inverse, and identity extend by generic equality and flatness. This gives a finite flat closed subgroup H of rank 3.

Its geometric special fiber is connected. Over the original unramified DVR, Oort–Tate parameters satisfy $ab = 3$ times a unit [50]. Connectedness gives $v(a) > 0$; since $v(3) = 1$, it follows that $v(a) = 1$ and b is a unit. Thus H^D is étale. Its generic fiber is the constant group $\mathbf{Z}/3\mathbf{Z}$; henselian uniqueness makes its integral model constant.

The original compact fine curve is proper, flat, and finitely presented, with universal $H^0(\mathcal{O}) = R_0$, by the regular model and multiplicity-one section just verified. Raynaud's Picard duality [39, 6.2.1] therefore turns $H \to \mathrm{Pic}$ into an H^D-torsor on the whole original fine curve. Rigidify at the cusp above ∞. The resulting torsor is finite étale.

Fix globally the based cover by quotienting the dual Jacobian by the subgroup generated by w, dualizing, and pulling back the isogeny along the ∞-based Abel–Jacobi map. Functoriality of Picard duality identifies its fine pullback with the same based generic torsor just constructed. At every place over 3 this étale model gives the integral branch in the original smooth fine cusp parameter. The conjugates of w are $\pm w$; transporting the full-four markings therefore gives this assertion at all such embeddings. At every other finite place, the two companion isogenies have differential composite [3]. Their formal maps on the original Néron models have integral inverse series at the identity. The based Abel–Jacobi map on the smooth Tate cusp chart then gives the same global integral branch. The fine parameter satisfies $q = \xi t^e$, with ξ a unit.

All hypotheses of Lemma 2.2 now hold: a primitive function integral over the affine j-line has its expansion in the normal Laurent cusp ring at every finite place, and the single based branch is fixed globally. That lemma makes the fine cover congruence. If Γ' is the original cover subgroup, the fine subgroup is $\Gamma' \cap \Gamma(4)$. It contains a principal congruence subgroup, hence so does Γ'. The compact cover is étale, so all cusp widths are unchanged. The compact-width argument of Theorem 2.3 then puts w in the Shimura subgroup.

An order-three diamond character has three-part of conductor at most 9. The natural diamond cover has split fiber at the generator cusp 0, and the cyclotomic character fiber at ∞. Twisting to split at ∞ makes its 0-fiber the cubic character field, up to inversion. The fine cusp above 0 has coefficient field unramified at 3, since only prime-to-three marking roots are needed. Our same étale torsor has unramified fiber there. A nontrivial conductor-nine cubic character is ramified at 3 and remains so after unramified extension. It is therefore absent. The remaining cubic character has conductor dividing $\mathrm{rad}(M) = \prod_{\ell \mid M} \ell$, as claimed. □

Lemma 8.14. *Let $f : X \to Y$ be a cyclic cover of degree 3 over $k \supset \mu_3$, with generator σ, and let $x_0 \in X(k)$ be a fixed point. If line bundles satisfy*

$$\sigma L - L = f^* s, \qquad s \in \mathrm{Pic}(Y)[3],$$

the cover associated to s, based at $f(x_0)$, splits at every branch point y whose single inverse image x has $k(x) = k(y)$.

Proof. Choose an isomorphism $\sigma L \simeq L \otimes f^* s$. Compose its three translates and choose a trivialization of s^3. The composite is a scalar automorphism of L. Rescale the chosen trivialization

so this scalar is 1. At a fixed point, choose fiber bases ℓ, m for L_x, s_y. If the isomorphism has scalar a in these bases, its three translates have scalar a^3 on that same fiber. The normalized trivialization sends m^3 to a^{-3}. The Kummer fiber is therefore split over $k(y)$. It splits at $f(x_0)$ in particular, so it is exactly the based cover. The common global normalization scalar has been fixed before evaluating any fiber. □

Lemma 8.15. *If a K-line bundle L on $X_0(N)$ satisfies*

$$(T-1)L = w$$

and $w \in \Sigma(N)[3](K)$ has conductor prime to 3, then $w = 0$. The same assertion holds with S.

Proof. The same tame character defines a compact étale Shimura cover at level $N/3$: that level is divisible by 9, so there is no cubic elliptic stabilizer, and every parabolic lower-right entry is 1 modulo each tame conductor prime. The matrix map for β sends

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \longmapsto \begin{pmatrix} a & 3b \\ c/3 & d \end{pmatrix},$$

so it preserves the character. Thus $w = \beta^* s$ for the same tame $s \in \Sigma(N/3)[3](K)$.

Take the two rational fixed cusps

$$x_0 = \infty, \quad x_1 = 1/3^n, \qquad y_0 = \infty, \quad y_1 = 1/3^{n-1}.$$

They have unchanged residue fields and full ramification degree 3. At the second cusp the exact matrix identity

$$\begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 3^n & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 3^{n-1} & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \tag{8.75}$$

gives $t_y = t_x^3$, with no translation constant. Lemma 8.14 makes the based s-cover split at both endpoints. On forgetting to level M, y_0 is ∞ and y_1 has denominator prime to M, hence is cusp 0. The ∞-based tame diamond fiber at y_1 is the cyclic cubic character field KF_χ. A nontrivial cyclic cubic field over $\mathbf{Q}$ cannot be contained in the quadratic field K. Thus $\chi = 0$ and $w = 0$. For S, use the Fricke-conjugate coordinate cover and the Fricke images of these two cusps. Fricke inverts the diamond character, which does not change its triviality. □

8.8. **The balanced operator.**

Proposition 8.16. *For every $P \in J_0(N)(\mathbf{Q})[3^\infty]$,*

$$DP = 0.$$

Proof. Set $R = DP$. By Proposition 8.12, all nine fine pullbacks of $S^i T^j R$ belong to the original Picard identity after the quadratic base change and have the same special class. Let $\mathcal{W}$ be the finite group generated by

$$S^i T^j (S-1)R, \qquad S^i T^j (T-1)R \qquad (0 \le i, j < 3).$$

Every element pulls back to the formal kernel of that original smooth group. The group is stable under S, T. It is also stable under τ, since

$$\tau R = S^{-1} T^{-1} R. \tag{8.76}$$

Lemma 8.2, and injectivity of the fine pullback on 3-primary torsion, give

$$3\mathcal{W} = 0, \qquad \tau|_{\mathcal{W}} = -1.$$

The relation $\tau S \tau = S^{-1}$ now gives $S = S^{-1}$ on $\mathcal{W}$; since $S^3 = 1$, it gives $S = 1$ there. Likewise $T = 1$ there. Writing $x = S - 1$, $y = T - 1$, we obtain the exact relations

$$x^2 R = xyR = y^2 R = 0, \qquad 3xR = 3yR = 0. \tag{8.77}$$

The group-ring identity $N_S D = 0$, $N_S = 1 + S + S^2$, therefore gives

$$0 = N_S R = 3R + 3xR + x^2 R = 3R.$$

The two remaining orbit differences $w_S = xR$, $w_T = yR$ are now anti-invariant three-torsion points with formal fine pullback. Lemma 8.13 puts both in the tame part of $\Sigma(N)[3]$. Apply Lemma 8.15 to the line bundle representing R, once for each coordinate. It gives

$$(S-1)R = (T-1)R = 0.$$

Equation (8.76) then makes R rational. We have proved that R is a rational point killed by 3, with original special class in $H^1(F, 1+\mathcal{J})$.

The rational generic fine point extends to the original unramified Néron model over R_0. After base change it is the section constructed above, by separatedness. Its special point is in the identity component, so it lies in the original Pic^0. Since its generic triple is zero and this group is separated, its triple is zero on the whole base. Rigidification at the cusp above ∞, together with $\mathrm{Pic}(R_0) = 0$, therefore supplies a compatible trivialization of the third power of the whole-model line bundle.

The fine curve is proper, flat, regular over the unramified DVR. Its geometric special fiber has $H^0(\mathcal{O}_F) = k$, by the original Picard comparison and its smooth multiplicity-one section. The outer component containing that cusp is a proper reduced connected component of multiplicity one; after the chosen unramified marking extension its constants are k. The ideal $\mathcal{J}$ of (8.74) has square zero, and is zero on this outer component away from its intersections. Thus Lemma 2.6 applies with $p = 3$, $r = 1$, and $I = \mathcal{J}$. It gives the integral based cube root in the same true fine cusp parameter.

Fix the original rational based cover associated to the cyclic subgroup generated by R. The whole-model trivialization just verified gives its fine branch at 3. At other primes the companion isogeny gives the integral formal inverse, as in Lemma 8.13. Lemma 2.2 and the same fine-subgroup intersection argument make the original cover congruence. Its unchanged compact cusp widths put the original point in $\Sigma(N)(\mathbf{Q})[3]$. This group is zero by the cyclotomic Galois action. Hence $R = 0$. □

Theorem 8.17. *For every $Q \in J_0(N)(K)[3^\infty]$,*

$$(1-S)(1-T)Q = 0. \tag{8.78}$$

Proof. By Proposition 8.16, it remains to treat $\tau Q = -Q$. The points

$$(T - T^{-1})Q, \qquad (S - S^{-1})Q$$

are rational. Apply Proposition 8.16 to both. On a finite 3-primary group,

$$T - T^{-1} = (T-1)(1+T^{-1}), \qquad (1+T^{-1})^{-1} = \frac{1 - T^{-1} + T^{-2}}{2},$$

and likewise for S. With $u = T - 1$, $v = S - 1$, we get

$$u^2 v Q = u v^2 Q = 0.$$

Thus $R = DQ$ is fixed by S, T. The identity $N_T D = 0$ gives $3R = N_T R = 0$, and conjugation gives $\tau R = -R$.

Proposition 8.12 extends the fine pullback of R to the base change of the original unramified Picard identity. Its reduction is fixed by τ, because that involution fixes the residue field and the original group. It is also negated by τ, and is killed by 3. Therefore its reduction is zero. Lemma 8.13 therefore puts R in the tame Shimura subgroup. Finally

$$R = (T-1)\big((S-1)Q\big).$$

Lemma 8.15 gives $R = 0$. Decomposition by the integer versions of $(1 \pm \tau)/2$ on the finite 3-primary group proves the theorem for arbitrary Q. □

8.9. **The corrected branch equation.**

Lemma 8.18. *Let $f : X \to Y$ be a cyclic cubic cover over $k \supset \mu_3$, with generator σ. Let $x_0, x_1 \in X(k)$ be fixed points with images y_0, y_1. Suppose line bundles and a divisor satisfy*

$$\sigma L = L \otimes f^*A \otimes \mathcal{O}_X(E), \qquad A^3 = \mathcal{O}_Y,$$

where E avoids x_0, x_1, and

$$(1 + \sigma + \sigma^2)E = \mathrm{Div}(h), \qquad h(x_0) = 1.$$

Then the fiber at y_1 of the A-cover split at y_0 has Kummer class $[h(x_1)]$ or its inverse in k^*/k^{*3}.

Proof. Choose a basis a_0 of A_{y_0} and a trivialization $\theta : A^3 \to \mathcal{O}_Y$ taking a_0^3 to 1. Choose the displayed bundle isomorphism and rescale it so that its fiber at x_0 is the identity in this basis and the canonical rational section 1 of $\mathcal{O}(E)$. Compose its three translates. After applying θ, the composite is an isomorphism

$$L \longrightarrow L \otimes \mathcal{O}_X((1 + \sigma + \sigma^2)E).$$

In the canonical rational section it is multiplication by c/h, since its divisor is $-\mathrm{Div}(h)$. Evaluation at x_0 makes $c = 1$.

At x_1, choose a basis a_1 of A_{y_1}. If the initial isomorphism has scalar b there, its three translates have the same scalar b on that fixed fiber. All translated divisors avoid x_1, so their canonical sections introduce no tangent character. Consequently

$$b^3\theta(a_1^3) = h(x_1)^{-1}.$$

This gives the asserted Kummer class, with inversion depending on the torsor convention. It has the same splitting field in either convention. □

Lemma 8.19. *If $h \in \mathcal{U}_N(K)$ satisfies $\tau h = h^{-1}$, $h(\infty) = 1$, and is regular and nonzero at $1/3^n$, then*

$$h(1/3^n) \in \mu_6.$$

Proof. Apply the full presentation (8.39) at level N. The infinity normalization gives leading coefficient 1 and infinity order zero. Dividing by its conjugate and using $\tau h = h^{-1}$ gives the exact identity

$$h^2 = \prod_{\substack{e|M \\ 0 \le j \le n-2}} R_{3^j e}^{d_{j,e}}. \tag{8.79}$$

The ordinary factors and every leading scalar cancel.

We compute the endpoint value of each factor. Let $m = 3^j e$, $H = n - j \ge 2$, $C = 3^H$. As $z \to 1/3^n$,

$$mz + k/3 \longrightarrow A_k/C, \qquad A_k = e + k3^{H-1}, \quad k = 1, 2.$$

Both fractions are in lowest terms with the same denominator C. Choose $\gamma_k = \begin{pmatrix} A_k & B_k \\ C & D_k \end{pmatrix} \in \mathrm{SL}_2(\mathbf{Z})$, and put $\delta = m(z - 1/3^n)$. The exact inverse formula is

$$\gamma_k^{-1}(A_k/C + \delta) = -\frac{D_k}{C} - \frac{1}{C^2\delta}. \tag{8.80}$$

The square-root factors in the two eta transformations are identical, since

$$C\gamma_k^{-1}(A_k/C + \delta) + D_k = -\frac{1}{C\delta}.$$

The leading exponential growth also agrees. Let $\varepsilon(\gamma)$ be the finite eta multiplier. Taking the ratio gives the finite nonzero limit

$$R_m(1/3^n) = e^{\pi i/36}\frac{\varepsilon(\gamma_1)}{\varepsilon(\gamma_2)}\exp\left(-\frac{\pi i(D_1 - D_2)}{12C}\right). \tag{8.81}$$

The first factor is the Euler normalization in

$$R_m = e^{\pi i/36}\frac{\eta(mz+1/3)}{\eta(mz+2/3)}.$$

All factors in (8.81) are roots of unity. The denominators and the local scale δ agree, so the amplitudes cancel.

Equation (8.79) now says that the square of $h(1/3^n)$ is a root of unity; hence the value itself is a root of unity. The cusp is rational and h is K-rational, so the value lies in K, whose roots of unity are exactly μ_6. □

Lemma 8.20. *For $n \geq 3$, the possible conductor-nine part of $\Sigma(3^nM)[3](K)$ is contained in $C_K(3^nM)_3$.*

Proof. At level 27, the eta coordinates

$$x = \frac{\eta(9z)^4}{\eta(3z)\eta(27z)^3}, \qquad Y = \frac{\eta(3z)^3}{\eta(27z)^3} \tag{8.82}$$

give, by (8.47),

$$x^3 = Y^3 g(3z) = Y^2 + 9Y + 27.$$

These are the modular coordinates of [13, §2.5]. Put $y = Y + 4$. Thus

$$X_0(27): \quad y^2 + y = x^3 - 7, \tag{8.83}$$

with infinity as origin, as the pole orders $2, 3$ in (8.82) show. Eta inversion at the cusp 0 gives

$$x(0) = \frac{\sqrt{3}\,27^{3/2}}{9^2} = 3, \qquad Y(0) = 0.$$

The exponential factors in x cancel; the factor in Y tends to zero. Thus $P_0 = (3, 4)$ is the rational cusp 0.

The tangent $y = 3x - 5$ has intersection equation

$$(3x-5)^2 + (3x-5) - (x^3 - 7) = -(x-3)^3.$$

Hence P_0 has order 3. From the eta translation law, the point action of the chosen cubic generator is $(x, y) \mapsto (\zeta x, y)$. The chord through $-P_0 = (3, -5)$ and $TP_0 = (3\zeta, 4)$ has slope $\zeta^2 - 1 = -(\zeta + 2)$ and intercept $1 + 3\zeta$. The group law consequently gives

$$(T-1)P_0 = (0, -2 - 3\zeta) =: u_0. \tag{8.84}$$

With the inverse pullback convention the result is $-u_0$, giving the same subgroup. It is nonzero, $\tau u_0 = -u_0$, and has order 3: the horizontal tangent at $x = 0$ has triple intersection there. It is represented by the difference of two K-cusps.

For diamond characters at level 27, $(\mathbf{Z}/27\mathbf{Z})^*$ has order 18; the parabolic condition kills $1 + 9\mathbf{Z}/27\mathbf{Z}$, of order 3; -1 kills the order-two factor. There are no additional cubic elliptic relations, since $d^2 + d + 1$ has no zero modulo 9, and no quadratic elliptic relation contributes. Thus $\Sigma(27) \simeq \mu_3$. The two points P_0, u_0 are independent, because their τ-characters are opposite, and they form an $\mathbf{F}_3$-basis of $J_0(27)[3](K)$. Its unique cyclotomic line is generated by u_0, so this is exactly $\Sigma(27)[3](K)$.

Every order-three character of $\mathbf{Z}_3^*$ factors modulo 9. At level $9M$ the parabolic relation already kills that factor. At depth $n \geq 3$, the possible surviving factor is the pullback of the character just exhibited at level 27. The forgetful degeneracy preserves the bottom-right entry modulo 9, so the upper character is represented by the pullback of the cusp divisor in (8.84). This proves the lemma. □

Proposition 8.21. *If $P \in J_0(N)(\mathbf{Q})[3^\infty]$, then*

$$(T-1)P \in C_K(N)_3 + \Sigma(N)[3](K) \quad \Longrightarrow \quad (T-1)P \in C_K(N)_3. \tag{8.85}$$

Proof. Absorb the wild part of the Shimura summand into the cusp class using Lemma 8.20. The remaining tame character comes from the adjacent level $N/3$, as in Lemma 8.15. We may therefore write

$$u = (T-1)P = \beta^* s + c, \qquad c \in C_K(N)_3,$$

with $s \in \Sigma(N/3)[3](K)$ tame. Put $w = \beta^* s$. Then $Tw = w$, $\tau w = -w$, and rationality of P gives

$$\tau(u-w) = -T^{-1}(u-w).$$

Choose a degree-zero K-cuspidal divisor E representing $u-w$, and set

$$E_{\mathrm{b}} = E - T\tau E.$$

It represents $2(u-w)$ and satisfies the exact divisor identity

$$\tau E_{\mathrm{b}} = -T^{-1} E_{\mathrm{b}}. \tag{8.86}$$

Its coefficient at each rational T-fixed cusp is zero. In particular it avoids $x_0 = \infty$ and $x_1 = 1/3^n$. Replace P, s by $2P, 2s$, and use E_{b} as E. Since 2 is invertible on the finite groups, this does not change whether $s = 0$.

Let $N_T = 1 + T + T^2$. We have $N_T u = 0$ and $N_T w = 3w = 0$. Thus $N_T E$ is principal as a K-divisor. Choose

$$h \in K(X_0(N))^*, \qquad \mathrm{Div}(h) = N_T E, \qquad h(x_0) = 1.$$

It is a cusp unit, regular and nonzero at both fixed cusps. Equation (8.86) gives $\mathrm{Div}(\tau h) = -\mathrm{Div}(h)$; the common normalization at x_0 then gives $\tau h = h^{-1}$.

Apply Lemma 8.18 with L representing P, A representing s, and this divisor E. The based fiber at $y_1 = 1/3^{n-1}$ has Kummer class $[h(x_1)]^{\pm 1}$. By Lemma 8.19, the value is in μ_6. Its class modulo cubes is therefore generated by ζ; if nontrivial, it defines exactly $K(\zeta_9)/K$, totally ramified above 3.

The same fiber of the tame diamond cover is KF_χ, where $F_\chi/\mathbf{Q}$ is the cyclic cubic field of conductor prime to 3. It is unramified above 3. These two descriptions force the Kummer class to be zero. Then F_χ would be contained in the quadratic field K, so $\chi = 0$ and $s = 0$. Restoring the absorbed wild cusp class proves (8.85). □

8.10. **Descent and induction.**

Lemma 8.22. *If $u \in J_0(N)(K)[3^\infty]$ is fixed by S, then*

$$u = \alpha^* R + c, \qquad R \in J_0(N/3)(K)[3^\infty], \quad c \in C_K(N)_3. \tag{8.87}$$

Proof. Choose a degree-zero line bundle L representing u. Its invariant class gives an isomorphism $SL \simeq L$. At the rational fixed cusp 0, let its fiber scalar be $a \in K^*$. The product of its three translates is a scalar on the proper connected curve, and evaluation at 0 makes it a^3. Dividing the isomorphism by a therefore gives a cyclic linearization.

All stabilizers occur at ramified cusps, by the absence of elliptic stabilizers proved after (8.62). Their tangent characters are primitive cube roots. Choose integral coefficients modulo 3 cancelling the fiber character at every ramified cusp. They can be chosen on K-closed cusps: $\mu_3 \subset K$, so the character and tangent character are constant on each such orbit. This gives a K-cuspidal divisor E such that $L(-E)$ has trivial stabilizer action everywhere. It descends to a line bundle M_0 on $X_0(N/3)$, with

$$L \simeq \alpha^* M_0 \otimes \mathcal{O}(E).$$

If $m = \deg M_0$, then $\deg E = -3m$. Using $\alpha^* 0 = 3(0)$, put

$$R = M_0(-m \cdot 0), \qquad E_0 = E + 3m \cdot 0.$$

Both have degree zero, and $u = \alpha^*[R] + [E_0]$. The cusp correction is torsion. If a kills the remaining pullback, then $\alpha_* \alpha^* = [3]$ gives $3a[R] = 0$. Thus $[R]$ is torsion. Project both terms to

their 3-primary parts, by multiplication by one integer on their common finite torsion group. This leaves a degree-zero cusp divisor and proves (8.87). □

Lemma 8.23. *For $n \geq 3$, both coordinate pullbacks agree on the order-three Shimura subgroup:*

$$\alpha^* s = \beta^* s \in \Sigma(N)[3](K) \quad (s \in \Sigma(N/3)[3](K)). \tag{8.88}$$

Moreover every $w \in \Sigma(N)[3](K)$ satisfies

$$Tw = w, \qquad Sw = w, \qquad \tau w = -w. \tag{8.89}$$

Proof. The matrix maps for the two degeneracies are

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \longmapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \qquad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \longmapsto \begin{pmatrix} a & 3b \\ c/3 & d \end{pmatrix}.$$

They preserve the bottom-right entry, hence give the same inflation of the compact diamond character. This proves (8.88) as an equality of line-bundle classes.

An order-three character at a prime different from 3 factors through that prime, since its higher principal units form a pro-ℓ group. At 3, it factors through modulus 9. Its conductor therefore divides $9\,\mathrm{rad}(M)$, which divides $N/3$ for $n \geq 3$. Direct conjugation gives

$$T \begin{pmatrix} a & b \\ c & d \end{pmatrix} T^{-1} = \begin{pmatrix} a + c/3 & b + (d-a)/3 - c/9 \\ c & d - c/3 \end{pmatrix}. \tag{8.90}$$

The lower-right entry changes by a multiple of $N/3$, so its character is unchanged. Conjugation by S changes d by $-(N/3)b$, with the same consequence. This proves the first two identities in (8.89). The last is the cyclotomic Galois action on Σ. It also shows that the K-rational 3-primary part of Σ is killed by 3, since K has no primitive ninth root of unity. □

Lemma 8.24. *If $Q \in J_0(N)(K)[3^\infty]$ and $(T-1)Q \in C_K(N)_3$, then*

$$Q = \beta^* R + c, \qquad R \in J_0(N/3)(K)[3^\infty], \quad c \in C_K(N)_3. \tag{8.91}$$

Proof. Choose a degree-zero K-divisor A representing Q and a degree-zero K-cuspidal divisor D_0 representing $(T-1)Q$. There is a function $f_0 \in K(X_0(N))^*$ with

$$(T-1)A - D_0 = \mathrm{Div}(f_0).$$

Its function norm $v = \mathrm{Nm}_\beta(f_0)$ has divisor $-\beta_* D_0$. Thus it is both a lower cusp unit and a function norm. Theorem 8.11 supplies $g \in \mathcal{U}_N(K)$ with $\mathrm{Nm}\, g = v$, with the constant unchanged. Cyclic Hilbert 90 gives $f_0/g = T(t)/t$ for a $t \in K(X_0(N))^*$. Hence $A' = A - \mathrm{Div}(t)$ satisfies

$$(T-1)A' = D_0 + \mathrm{Div}(g),$$

which is supported on cusps.

The noncuspidal part of A' is an invariant integral divisor. The cover is unramified there, so each point above a given closed point has the same coefficient and ramification index one. This coefficient descends unchanged, including at inert points. Thus the noncuspidal part is the pullback of an integral divisor B downstairs. We obtain

$$A' = \beta^* B + E$$

with E a K-cuspidal divisor. If $m = \deg B$, use $\beta^*\infty = 3\infty$ to replace B, E by

$$B - m\infty, \qquad E + 3m\infty.$$

They now both have degree zero. Their classes satisfy $Q = \beta^* R + c$. Manin–Drinfeld makes c torsion, and $\beta_*\beta^* = [3]$ then proves that R is torsion. Simultaneous integer primary projection gives (8.91). □

Proof of Theorem 8.1. We prove (8.1) for the entire K-group by induction on n. The base $n = 2$ is Corollary 8.9. Assume it at $N/3 = 3^{n-1}M$, where $n \geq 3$. Take a point Q first in either of the two τ-eigenspaces, and put $u = (T-1)Q$. By Theorem 8.17, $(S-1)u = 0$. Lemma 8.22, the full lower induction hypothesis, and Lemma 8.23 give

$$u = C + w, \qquad C \in C_K(N)_3, \quad w \in \Sigma(N)[3](K). \tag{8.92}$$

If $\tau Q = Q$, the point is rational and Proposition 8.21 gives $u \in C_K(N)_3$. If $\tau Q = -Q$, then

$$\tau u = (T^{-1} - 1)(-Q) = T^{-1}u.$$

Applying this and (8.89) to (8.92) gives the exact equality

$$2w = \tau C - T^{-1}C \in C_K(N)_3. \tag{8.93}$$

The group of K-cuspidal classes is stable under both operations. Since $3w = 0$, the integer identity $w = 2(2w)$ puts w itself in that cuspidal group. Thus u is cuspidal in the anti case as well.

In either case Lemma 8.24 gives $Q = \beta^* R + c$. The lower point R is an arbitrary K-rational 3-primary torsion point. The induction hypothesis therefore applies to it: $R = c' + s'$, with $c' \in C_K(N/3)_3$ and $s' \in \Sigma(N/3)[3](K)$. Consequently

$$Q = (c + \beta^* c') + \beta^* s' \in C_K(N)_3 + \Sigma(N)[3](K).$$

The two projectors $(1 \pm \tau)/2$ now give this containment for every Q, and the reverse containment is immediate. This proves (8.1) for every $n \geq 2$.

Finally let $P \in J_0(3^n M)(\mathbf{Q})[3^\infty]$. Write $P = c + w$ by the proved full-K statement, and choose a degree-zero K-cuspidal divisor E representing c. Since $\tau w = -w$,

$$2P = c + \tau c.$$

This last class is represented by the rational divisor $\mathrm{Tr}_{K/\mathbf{Q}} E$; its base change is $E + \tau E$, and its degree is zero. If $3^a P = 0$, choose an integer b with $2b \equiv 1 \pmod{3^a}$. Then $b \,\mathrm{Tr}_{K/\mathbf{Q}} E$ is a rational degree-zero cusp divisor representing P. Proposition 8.4 supplies the remaining levels with $v_3(N) \leq 1$. The reverse inclusion at every level is Manin–Drinfeld. □

9. The primary parts at primes at least five

Fix a prime $p \geq 5$ and write $N = p^n M$, where $(p, M) = 1$. Throughout this section

$$K = \mathbf{Q}(\zeta_p), \qquad h = (p-1)/2,$$

and F is the quadratic subfield of K. A cuspidal class over a field means a class represented by a degree-zero cuspidal divisor over that field. We shall establish

Theorem 9.1. *For every positive integer N and every prime $p \geq 5$,*

$$J_0(N)(\mathbf{Q})[p^\infty] = C_{\mathrm{rat}}(N)[p^\infty].$$

Theorem 2.1 covers $n = 0, 1$. The proof for $n \geq 2$ uses a marked tower. We induct on its scalar-invariant K-rational p-primary torsion group.

9.1. **The marked arithmetic tower.** For $n \geq 2$ set

$$\Gamma_n = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma_0(p^n M) : a = d = \pm 1 \pmod{p} \right\}.$$

Let Y_n be the corresponding compact curve with its marked $\mathbf{Q}$-model. Its elliptic moduli datum is (E, C, P), where C is cyclic of order $p^n M$ and P, modulo sign, generates the standard order-p subgroup of C. The standard subgroup here is the Katz–Mazur subgroup [25, 6.7.2], defined on the whole fiber. The forgetful map

$$\pi_n : Y_n \longrightarrow X_0(p^n M)$$

has degree h.

Here is an intrinsic specification of the arithmetic scalar action. Use the principal modular-function field of level N, with constants $\mathbf{Q}(\zeta_N)$, and the right convention $f|\gamma(z) = f(\gamma z)$. The arithmetic groups fixing $K(Y_n)$ and its scalar-invariant subfield are, respectively,

$$H_K = \{g \in \mathrm{GL}_2(\mathbf{Z}/N) : c = 0,\ a = d = \pm 1 \pmod p\}, \tag{9.1}$$

$$H_Z = \{g \in \mathrm{GL}_2(\mathbf{Z}/N) : c = 0,\ a = d \pmod p\}. \tag{9.2}$$

This is the modular-function Galois description [58, Proposition 3.1], with both groups transposed if its left convention is used. The determinant image of H_K is the subgroup congruent to 1 modulo p, whereas that of H_Z is the square subgroup modulo p. Their fixed constants are consequently K and F.

Choose compatible Teichmüller scalars at p and the scalar 1 at every prime dividing M. They give

$$Z = H_Z/H_K, \qquad |Z| = h.$$

This group acts semilinearly on Y_n/K and faithfully on K/F. If the $\mathbf{Q}$-defined diamond δ_a sends P to $a^{-1}P$, the point action is

$$z_a = \delta_a \sigma_{a^2}, \qquad \sigma_c(\zeta_p) = \zeta_p^c. \tag{9.3}$$

One can check the exponent directly at $n = 2$. Passing to the middle elliptic curve changes the marking to a mixed symplectic frame $(i : \mu_p \to E[p], Q : \mathbf{Z}/p \to E[p])$. Its constant frame over K is $\phi = (i(\zeta_p), Q)$, and

$$\phi(\delta_a \sigma_c x) = (\sigma_c \phi(x)) \,\mathrm{diag}(c^{-1}a, a^{-1}).$$

For $c = a^2$ this matrix is the scalar $a^{-1}I$. The inverse in the first entry occurs because $\sigma_c(i(\zeta_p)) = (\sigma_c i)(\zeta_p^c)$.

Put

$$A_n = \mathrm{Jac}(Y_n)(K)[p^\infty]^Z, \qquad C_n = C_K(Y_n)[p^\infty]. \tag{9.4}$$

The group C_n is finite by Manin–Drinfeld. Our induction target is $A_n \subset C_n$. Pullback of a point of $J_0(p^n M)(\mathbf{Q})[p^\infty]$ belongs to A_n, since it is fixed by both diamonds and arithmetic Galois.

For $n \geq 3$ define

$$S = \begin{pmatrix} 1 & 0 \\ -N/p & 1 \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 1/p \\ 0 & 1 \end{pmatrix}. \tag{9.5}$$

Both normalize Γ_n and have order p on Y_n. For example, conjugation by T changes the upper-right entry of γ to $b + (d-a)/p - c/p^2$, which is integral, and preserves the other required congruences. Conjugation by S changes the lower-left entry to $c - (N/p)(a-d) - (N^2/p^2)b$, divisible by N. Their commutator is

$$STS^{-1}T^{-1} = \begin{pmatrix} 1 + N/p^2 & -N/p^3 \\ -N^2/p^3 & 1 - N/p^2 + N^2/p^4 \end{pmatrix} \in \Gamma_n. \tag{9.6}$$

All entries and congruences in this display are valid already at $n = 3$. In particular S, T commute as automorphisms of Y_n.

Their two quotients are the lower marked curve:

$$\alpha : Y_n/\langle S\rangle \xrightarrow{\sim} Y_{n-1}, \quad z \longmapsto z, \qquad \beta : Y_n/\langle T\rangle \xrightarrow{\sim} Y_{n-1}, \quad z \longmapsto pz. \tag{9.7}$$

Indeed the p cosets of Γ_n in Γ_{n-1} are distinguished by $c/(N/p)$ modulo p; an S-power removes that residue. Conjugating Γ_n by $\mathrm{diag}(p, 1)$ makes its upper-right entry divisible by p, and adjoining the upper unipotents gives Γ_{n-1}. For the marking in β, let $\phi : E \to E/C_p$ be the first isogeny. The old P identifies $\ker \phi^\vee$ with μ_p. The next standard p-line is transverse to this kernel; pairing with it identifies that line with $\mathbf{Z}/p$ and selects its new generator modulo sign. Replacing P by rP multiplies the identification with the dual kernel by r^{-1} and the selected

generator by r. Thus both maps commute with the same diamonds. They are Z-equivariant also directly from (9.1)–(9.2), because scalar matrices commute with I and $\mathrm{diag}(p,1)$.

The lower group has no elliptic stabilizers: its trace is ± 2 modulo p, whereas an elliptic element of $\mathrm{SL}_2(\mathbf{Z})$ has trace 0 or ± 1. Hence the maps (9.7) are unramified away from cusps. The cusp 0 is K-rational and totally ramified for α. For β, the marked cusp ∞ becomes rational over K, is the unique cusp above the selected lower cusp, and has the parameters

$$q = t^p. \tag{9.8}$$

9.2. **A saturated quadratic descent at the lower level.** We first prove the arithmetic descent needed for the seed.

Lemma 9.2. *If $(p,M)=1$ and F is the quadratic subfield of $\mathbf{Q}(\zeta_p)$, then*

$$J_0(pM)(F)[p^\infty] = J_0(pM)(\mathbf{Q})[p^\infty].$$

Proof. Work first over $R=\mathbf{Z}_p$, with fraction field $K_0=\mathbf{Q}_p$, and write $B=J_0(pM)_{K_0}$ and $\mathcal{B}$ for its Néron model. The variety is semistable. Let

$$\mathscr{F}[p^a] = (\mathcal{B}^0[p^a])^f$$

be the finite part. These groups form the finite-identity p-divisible group $\mathscr{F}$; let $\mathscr{C}=\mathscr{F}^0$ be its connected part. Each $\mathscr{C}[p^a] \hookrightarrow \mathcal{B}^0[p^a]$ is a finite flat closed subgroup with connected special fiber. The finite-part theorem and its connected–étale sequence give

$$T = T_pB, \qquad T_f = T_p\mathscr{F}_{K_0}, \qquad T_c = T_p\mathscr{C}_{K_0},$$
$$T_f/T_c = T_p\mathscr{F}^{\mathrm{et}}_{K_0}, \qquad T/T_f = \Upsilon \otimes \mathbf{Z}_p,$$

where Υ is an unramified lattice [21, IX, 2.2.3, 5.1, (5.5.8), and Proposition 5.6]. Both quotients are free. Thus $E=T/T_c$ is free and T_c is saturated. Tensoring with $\mathbf{Q}_p/\mathbf{Z}_p$ consequently gives the exact sequence

$$0 \longrightarrow \mathscr{C}_{K_0}[p^\infty](\overline{K}_0) \longrightarrow B[p^\infty](\overline{K}_0) \longrightarrow E\otimes \mathbf{Q}_p/\mathbf{Z}_p \longrightarrow 0. \tag{9.9}$$

The construction $B \mapsto E\otimes\mathbf{Q}_p$ is additive and isogeny invariant. Homomorphisms extend to Néron models and preserve both the finite part and its connected part; an isogeny and a companion have scalar composites. For a good abelian variety, E is the Tate module of the étale quotient of its p-divisible group, and is unramified. For a purely toric semistable variety the finite part is connected, and E is its unramified period lattice. The old part of $J_0(pM)$ is isogenous to $J_0(M)^2$, of good reduction at p, and the p-new quotient is purely toric [11, §7.1 and Lemma 8.3], since $p\|pM$. Hence $E\otimes\mathbf{Q}_p$ is unramified. The free lattice E embeds in this rational representation, so E is unramified as well.

The unique completion L of F at p is totally ramified quadratic over K_0. Its nontrivial automorphism is the restriction of some $\sigma \in I_{K_0}$. For $P\in B(L)[p^\infty]$, sequence (9.9) puts $\delta=\sigma P - P$ in $\mathscr{C}_{K_0}[p^\infty](L)$. For some a it is a point of the finite proper $\mathscr{C}[p^a]$, so it extends over $\mathcal{O}_L$ to the base change of the *original* $\mathcal{B}^0$. Its special point is zero: a finite connected group over a perfect field has only its identity as a geometric point.

There is no nonzero p-primary torsion in that formal kernel. Normalize v_L by $v_L(\varpi)=1$, so $v_L(p)=2$. In smooth formal coordinates,

$$[p](X) = pX + p\,(\text{terms of degrees } 2,\dots,p-1) + (\text{terms of degree at least } p).$$

The assertion about the intermediate coefficients follows by reducing the homomorphism $[p]$ in characteristic p: its differential vanishes identically, so its positive monomials below degree p vanish. If the minimum coordinate valuation of a nonzero point is $a\ge 1$, the linear term in a minimal coordinate has valuation $2+a<pa$, strictly below the other terms. Its p-multiple is nonzero with depth exactly $a+2$, and iteration gives the same conclusion at every further multiple. Thus $\delta=0$. The local nontrivial automorphism is the global nontrivial automorphism of $F/\mathbf{Q}$,

and an embedding of fields gives an injection on points. Therefore every $P \in J_0(pM)(F)[p^\infty]$ is rational over $\mathbf{Q}$. □

9.3. **The integral full-p seed.** Conjugation by $\mathrm{diag}(p,1)$ identifies Y_2/K with the full-p curve with tame cyclic M-structure. Its geometric group is $G = \mathrm{PSL}_2(\mathbf{F}_p)$. Let $\Omega = \mathbf{P}^1(\mathbf{F}_p)$ and let B_L be the stabilizer of $L \in \Omega$. For a subgroup U, write $N_U = \sum_{u\in U} u$. Define

$$H = \sum_{L\in\Omega} N_{B_L} - N_G, \qquad u = (p^2-1)/2. \tag{9.10}$$

Write $\rho(g) = \#\Omega^g$, so the coefficient of g in H is $\rho(g)-1$. The action is doubly transitive: the stabilizer of ∞ contains all translations on $\mathbf{F}_p$. Also $|G| = p(p^2-1)/2 = pu$, by choosing a nonzero first column and then one of the p second columns of determinant one, and quotienting by $\{\pm 1\}$. Double-counting fixed pairs gives $\sum_{h\in G}\rho(h) = |G|$. For fixed g, the sum $\sum_h \rho(h)\rho(h^{-1}g)$ counts triples (h,x,y) with $hx = x$ and $hy = gy$. The pairs with $x = y$ contribute $\rho(g)|G|/(p+1)$. For $x \neq y$, a solution exists precisely when $x \neq gy$, and then there are $|G|/(p(p+1))$ solutions. There are $p(p+1) - (p+1-\rho(g)) = p^2 - 1 + \rho(g)$ such pairs. Consequently

$$\sum_{h\in G}\rho(h)\rho(h^{-1}g) = \frac{|G|}{p+1}\rho(g) + \frac{|G|}{p(p+1)}(p^2-1+\rho(g)) = |G| + u(\rho(g)-1).$$

Subtracting $|G|$ gives the coefficient of g in H^2. Thus, in $\mathbf{Z}[G]$,

$$H^2 = uH, \qquad \mathrm{aug}(H) = 0, \qquad p \nmid u. \tag{9.11}$$

In particular $e = H/u$ is a $\mathbf{Z}_p[G]$-projector. On the permutation module $\mathbf{Z}_p^\Omega$, the (x,y) entry of H is $\sum_{gy=x}(\rho(g)-1)$. For $x = y$, the sum of ρ over B_x is $2|B_x|$, since B_x has two orbits on Ω. Thus the diagonal entry is $|B_x| = p(p-1)/2$. For $x \neq y$, a fixed point z of such a g must lie outside $\{x,y\}$. Each of the $p-1$ choices of z contributes $|G|/(p(p+1)) = (p-1)/2$ elements. The off-diagonal entry is therefore

$$\frac{(p-1)^2}{2} - \frac{p(p-1)}{2} = -\frac{p-1}{2}.$$

Hence H acts as zero on constants and as u on the augmentation submodule.

Let $A = \mathrm{Jac}(Y_2)$ and set $C = \mathrm{im}(u-H)$ and $D = \mathrm{im}\, H$. There are homomorphisms

$$i : C \longrightarrow A, \qquad r : A \longrightarrow C, \qquad ir = u - H, \quad ri = [u], \quad Hi = 0. \tag{9.12}$$

Indeed $(u-H)^2 = u(u-H)$ and $H(u-H) = 0$, which give $ri = [u]$ and $Hi = 0$. An element of $C \cap D$ is killed by u, since H acts as zero on C and as u on D. The map $C \times D \to A$ is therefore an isogeny with kernel killed by u: it is surjective since $[u]A = (u-H)A + HA \subset C + D$, and its kernel consists of the pairs $(x,-x)$ with $x \in C \cap D$. It induces an isomorphism on p-divisible groups and a split comparison of the formal Lie lattices.

For geometry at p, add full level 3, fix its Weil pairing, and choose a geometrically connected component of that pairing locus. We transport this component through the degeneracy maps. The tame problem $[\Gamma(3)] \times [\Gamma_0(M)]$ is representable and finite étale over the elliptic stack at p. This remains true when $3 \mid M$. The forgetting degree divides 24 and is a p-unit. On the fine curve the group is $\widetilde{G} = \mathrm{SL}_2(\mathbf{F}_p)$ and $\widetilde{u} = p^2 - 1$. On central-even pullback, $\widetilde{H} = 2H$ and $\widetilde{u} = 2u$, so the projectors agree. We use the fine curve for the following integral calculations and return to the coarse curve before taking Borel quotients.

The full-p semistable model for this representable tame problem exists after the tame extension with absolute ramification

$$e_0 = p^2 - 1.$$

Its components and actions are described in [20, Theorem 2.1, Remark 2.2, and §§2.2.1–2.2.4]. Let SS be the set of supersingular points of the chosen tame component. There is an Igusa

component I_L for every $L \in \Omega$, and a Drinfeld component D_s for every $s \in \mathrm{SS}$. Each D_s meets each I_L once. Its equation over $k = \overline{\mathbf{F}}_p$ is

$$\alpha^p\beta - \alpha\beta^p = c_s, \qquad c_s \neq 0. \tag{9.13}$$

The nodes have thickness one. This model is semistable for every $p \geq 5$.

The graph cycle lattice is

$$\mathbf{Z}[\Omega]_0 \otimes \mathbf{Z}[\mathrm{SS}]_0. \tag{9.14}$$

Indeed, write a cycle as an integer matrix on the edges; the cycle equations are that every row and every column sum is zero. The full-p group acts on Ω and fixes the tame supersingular label. Thus H is multiplication by u on the torus character lattice of the special Picard variety. It follows from (9.12) that the torus of C is zero. Semistability and reduction ranks are invariant under isogeny and additive for products [21, IX, 2.2.6–2.2.9 and §12]; hence the complement C has good reduction over this extension.

Let $\mathcal{A}, \mathcal{C}$ denote the Néron models there. The maps in (9.12) extend and give $\mathcal{C} \to \mathcal{A}^0 \to \mathcal{C}$ with composite $[u]$. In particular di is a primitive split injection of smooth Lie lattices. If

$$B = \prod_{L\in\Omega} \mathrm{Jac}(I_L) \times \prod_s \mathrm{Jac}(D_s),$$

the normalization sequence is $0 \to T_0 \to \mathcal{A}^0_k \xrightarrow{\nu} B \to 0$. The map r_k kills the torus, since a torus has no nonzero homomorphism to an abelian variety. It therefore factors as $\eta : B \to \mathcal{C}_k$. Set $\rho = \nu i_k$. Then

$$\eta\rho = [u], \qquad \rho\eta = (u - H)_B. \tag{9.15}$$

These identities prove

$$\mathcal{C}_k[p^\infty] \simeq (1-e)B[p^\infty], \qquad \mathrm{Lie}\,\mathcal{C}_k \simeq (1-e)\,\mathrm{Lie}\,B. \tag{9.16}$$

The inverse uses the p-unit u. The identification of $\mathcal{A}^0$ with the Picard identity component uses the regular reduced model and [39, Theorems 8.1.4 and 8.2.1].

9.4. **The two surviving formal depths.** We now calculate the congruence filtration of $\mathcal{C}$. Work over a strictly henselian complete DVR $\mathcal{O}$ with residue field k, uniformizer ϖ, and $v(p) = e_0 = p^2 - 1$. One may take $\varpi = (1-\zeta_p)^{1/(p+1)}$ after an unramified extension. Use the right action $r(g)$ on frames and its left pullback action on Picard groups. In the full-p model, the raw inertia action $\gamma(v)$ satisfies $\gamma(v)(\varpi) \equiv v\varpi \pmod{\varpi^2}$. If $v^{p+1} = 1$, it fixes K, is the identity on Igusa components, and sends

$$(\alpha, \beta) \longmapsto (v^{-1}\alpha, v^{-1}\beta)$$

on every Drinfeld component. For $\lambda \in \mathbf{F}_p^\times$ the combination

$$z_\lambda = r(\lambda I)\gamma(\lambda)$$

preserves the pairing component. It is the intrinsic scalar of (9.3): the constant-field action is the square λ^2. It fixes the reduced Drinfeld coordinates and acts on an Igusa generator by $P_I \mapsto \lambda P_I$. Its leading uniformizer character is λ. These assertions follow by substituting $\alpha = x/\varpi$, $\beta = y/\varpi$ in the frame action; they are the two commuting actions of [20, §§2.2.3–2.2.4].

Let $Q \in C(K)[p^\infty]$ be scalar-fixed and set $V = \mathbf{Z}_p[G]Q$. Every element of V is fixed by these two tame actions. The assertion on the fine auxiliary curve includes $\lambda = -1$, since the pullback is central-even. For $a \geq 1$ let

$$\mathcal{C}^a(\mathcal{O}) = \ker\bigl(\mathcal{C}(\mathcal{O}) \to \mathcal{C}(\mathcal{O}/\varpi^a)\bigr).$$

Smoothness gives the canonical quotient

$$\mathcal{C}^a/\mathcal{C}^{a+1} \simeq \mathrm{Lie}\,\mathcal{C}_k \otimes_k (\varpi^a)/(\varpi^{a+1}). \tag{9.17}$$

Semilinear pullback acts on both factors. For example, on transition functions $1+\epsilon f_{ij}$ it acts on the $H^1(\mathcal{O})$ coefficient and on ϵ; a uniformizer character λ therefore contributes λ^a.

The formal multiplication estimate is

$$\operatorname{depth}([p]x) \geq \min(e_0 + a, pa) \tag{9.18}$$

for a point of depth a. If $a > e_0/(p-1) = p+1$, the linear term in a coordinate of minimal valuation has strictly smaller valuation than all higher terms. Its multiple is nonzero of depth $e_0 + a$, and iteration proves that no nonzero p-primary torsion has depth greater than $p+1$.

Plane adjunction on (9.13) gives the basis

$$\alpha^r \beta^s \omega_0, \qquad \omega_0 = d\alpha/\alpha^p, \qquad r, s \geq 0, \quad r+s \leq p-2$$

of holomorphic differentials on a tail. Writing $d = r+s$, its $\gamma(v)$-character is v^{p-1-d}; the dual $H^1(\mathcal{O})$ character is $v^{d-(p-1)}$. The scalar acts trivially. The degree-zero differential is full-p invariant. Indeed, differentiating the equation gives $\alpha^p\, d\beta = \beta^p\, d\alpha$, and for $\alpha' = a\alpha + c\beta$ with $a, c \in \mathbf{F}_p$,

$$\frac{d\alpha'}{(\alpha')^p} = \frac{(a + c\beta^p/\alpha^p)d\alpha}{a\alpha^p + c\beta^p} = \frac{d\alpha}{\alpha^p}.$$

The tails also have p-rank zero. To see this directly, use

$$\beta = (\beta/\alpha)^p \alpha + (c_s^{1/p}/\alpha)^p.$$

After substitution in $\alpha^r \beta^s\, d\alpha/\alpha^p$, each term is a p-th power times $\alpha^j\, d\alpha$ with $0 \leq j \leq p-2$, so Cartier kills every holomorphic differential. This gives the claimed p-rank.

On the Igusa component with kernel line $\mathbf{F}_p e_1$, the ordinary frame is $(0, P_I)$. An upper Borel element $\left(\begin{smallmatrix} t & b \\ 0 & t^{-1} \end{smallmatrix}\right)$ sends its surviving generator to $t^{-1}P_I$. The unipotent radical is consequently the identity on the whole smooth component, by equality on the dense ordinary open. If a differential has diamond exponent b, its dual $H^1(\mathcal{O})$ exponent is $-b$, and its Borel character is t^b. Combining this with (9.17) gives

component at depth a	z_λ	$\gamma(v),\ v^{p+1}=1$
tail, homogeneous degree d	λ^a	$v^{a+d-(p-1)}$
Igusa, differential exponent b	λ^{a-b}	v^a

(9.19)

for the tangent contribution. Within $1 \leq a \leq p+1$, a fixed tail term requires $a = 0 \pmod{p-1}$ and $a+d-(p-1) = 0 \pmod{p+1}$. Since $p+1 < 2(p-1)$, its only possibility is $a = p-1, d = 0$. A fixed Igusa term requires $a = 0 \pmod{p+1}$ and $a = b \pmod{p-1}$, hence $a = p+1, b = 2$. Fix $L = \mathbf{F}_p(1,0)$ and $B = B_L$. Write $k(t^2)$ for the one-dimensional B-module on which $\left(\begin{smallmatrix} t & * \\ 0 & t^{-1} \end{smallmatrix}\right)$ acts by t^2. For the diamond-exponent -2 eigenspace set

$$\mathcal{R}_{\mathrm{I}} = \operatorname{Hom}_B(k(t^2), H^1(I_L, \mathcal{O})_{\diamond,-2}),$$

where the subscript means that the diamond s acts as s^{-2}. The preceding Borel calculation identifies this eigenspace with $k(t^2) \otimes_k \mathcal{R}_{\mathrm{I}}$; we give the multiplicity space $\mathcal{R}_{\mathrm{I}}$ the trivial G-action. The two layers are therefore

depth	source	full-p module
$p-1$	tail degree zero	trivial
$p+1$	Igusa diamond exponent -2	$\operatorname{Ind}_B^G(t^2) \otimes_k \mathcal{R}_{\mathrm{I}}$.

(9.20)

Set

$$V_0 = V \cap \mathcal{C}^1(\mathcal{O}), \qquad I_0 = V \cap \mathcal{C}^{p+1}(\mathcal{O}).$$

The leading-term calculation proves $V_0 \subset \mathcal{C}^{p-1}$, $V_0 \cap \mathcal{C}^p = I_0$, and $V \cap \mathcal{C}^{p+2} = 0$. Thus V_0/I_0 is G-trivial and there is an injective $\mathbf{F}_p[G]$-map

$$I_0 \hookrightarrow \operatorname{Ind}_B^G(t^2) \otimes_k \mathcal{R}_{\mathrm{I}}. \tag{9.21}$$

Equation (9.18) sends a nonzero p-multiple of depth at least $p-1$ to depth at least $p(p-1) > p+1$. Hence

$$pV_0 = 0. \tag{9.22}$$

The nonformal quotient V/V_0 is also G-trivial. The torus has no geometric p-power points, and the tails have p-rank zero, so reduction injects into the Igusa point modules in (9.16). Scalar invariance selects invariants under the diamond group $D_I = \mathbf{F}_p^\times$. Its quotient $q : I_L \to I_L/D_I$ has degree $p-1$ and satisfies

$$q_* q^* = [p-1], \qquad q^* q_* = \sum_{\delta \in D_I} \delta.$$

It identifies the invariant p-primary points with the pullback of the quotient Jacobian's points. The possible Frobenius factor of the exotic Igusa map to the tame base is retained: only $p-1$ has been inverted. The quotient coefficient groups on the different components are canonically identified over the tame base, and their residual maps over that radicial map are the identity. Thus the full-p module is a permutation module on Ω with trivial coefficient action. Constants split from augmentation using the p-unit $p+1$. The operator H is u on augmentation and zero on constants. Since H kills C, only constants remain in the reduction of V.

Let $I_G = \ker(\mathrm{aug} : \mathbf{Z}_p[G] \to \mathbf{Z}_p)$ be the augmentation ideal. On V/I_0, every $g-1$ maps into V_0/I_0 and kills that subgroup. Thus $(g-1)(g'-1) = 0$ and $g \mapsto g-1$ is a homomorphism to the abelian group $\mathrm{Hom}(V/V_0, V_0/I_0)$. Put $U(x) = \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}$ for $x \in \mathbf{F}_p$. Both $\mathrm{SL}_2(\mathbf{F}_p)$ and $\mathrm{PSL}_2(\mathbf{F}_p)$ are perfect for $p \geq 5$: choose $t^2 \neq 1$ and use

$$[\mathrm{diag}(t, t^{-1}), U(x)] = U((t^2-1)x)$$

and the analogous lower-root formula. The two root subgroups generate $\mathrm{SL}_2(\mathbf{F}_p)$. Consequently

$$I_G V \subseteq I_0. \tag{9.23}$$

Finally the module in (9.21) has no G-invariants. A fixed section on the transitive component set is determined by a value fixed by the Borel. The Borel torus acts on that value by t^2, nontrivially for every $p \geq 5$, so it is zero.

9.5. **Primitive depth and one global compact cover.** We show that every element of $I_G V$ is G-fixed. Take $0 \neq w \in I_G V$. By (9.23) and (9.22), it has order p and exact depth $p+1$ in the good abelian scheme $\mathcal{C}$. Its cyclic schematic closure $\mathcal{H}_w \subset \mathcal{C}[p]$ is finite flat of rank p. Tate–Oort's local description gives

$$\mathcal{H}_w = \mathrm{Spec}\, \mathcal{O}[x]/(x^p - ax), \qquad ab = p\varepsilon, \quad \varepsilon \in \mathcal{O}^\times, \tag{9.24}$$

with duality exchanging a, b [50, Theorem 2 and Remarks 1, 2, 5, 6]. Since w is formal, $v(x(w)) = v(a)/(p-1) > 0$.

For smooth parameters $t_1, \ldots, t_g$ of $\mathcal{C}$, write their restrictions uniquely as

$$t_i|_{\mathcal{H}_w} = \sum_{j=1}^{p-1} c_{ij} x^j, \qquad c_{ij} \in \mathcal{O}.$$

The special closed immersion induces a surjection on cotangent spaces. Its target is generated by x, so some c_{i1} is a unit. For this i the linear term has strictly smaller valuation than every higher term. Every other coordinate has valuation at least $v(x(w))$. Consequently

$$\mathrm{depth}_{\mathcal{C}}(w) = v(x(w)) = v(a)/(p-1).$$

Since the depth is $p+1 = e_0/(p-1)$, $v(a) = e_0$ and b is a unit. Thus $\mathcal{H}_w^D$ is étale, and over the strictly henselian base $\mathcal{H}_w \simeq \mu_p$.

The map to the Picard scheme is also a closed immersion:

$$\mathcal{H}_w \longrightarrow \mathcal{C} \xrightarrow{i} \mathcal{A}^0 = \mathrm{Pic}^0_{\mathscr{X}/\mathcal{O}}\,.$$

Its scheme-theoretic kernel is zero by $ri = [u]$ and $p \nmid u$. A finite monomorphism into this separated group is a closed immersion. Raynaud's Picard–torsor duality [39, Propositions 6.2.1 and 6.2.3] therefore produces a torsor under the étale group $\mathcal{H}_w^D$ on the complete semistable curve, including its nodes and cusps.

To apply a global denominator statement, fix the cover before choosing any place. Over one number field L containing the fine curve, w, and a chosen cusp c, form $f : A \to A/\langle w\rangle$. Pull its dual back along the Abel–Jacobi map based at c. The identity over its zero value gives a based generic cover

$$X_w \longrightarrow \mathscr{X}_L.$$

It is geometrically connected: equivalently, the line bundle of exact order p defining this based Kummer cover has no smaller order. The local Raynaud torsor has the same generic Picard class. Rigidification at c removes the torsor from the base, so its generic cover is this same X_w. The existence of the section also removes the relative cohomological obstruction: pullback on base cohomology has the section as a retraction, and its possible kernel is zero.

Every place of L above p is covered by the depth argument. Indeed arithmetic conjugation normalizes the full-p group and the intrinsic scalar subgroup, and preserves H. For any conjugation τ, the point τQ is again scalar-fixed in the complementary factor and

$$\tau(I_G V) \subseteq I_G \mathbf{Z}_p[G](\tau Q).$$

The preceding argument, with the transported tame component and cusp, applies to each conjugate.

At a place away from p, a companion to $f^\vee$ has composite $[p]$ with it. The two integral Néron Lie matrices have product the unit scalar p, so both are invertible. The based Abel–Jacobi map extends on the smooth cusp neighborhood to the original Néron model. Its lift under the formal inverse of $f^\vee$ gives an integral branch of the same based X_w.

The cusp parameters are as follows. At the full-p, full-3 cusp the parameter is $t = q^{1/(3p)}$, and without the auxiliary marking it is $q^{1/p}$; the canonical cyclic M-subgroup is μ_M and does not enlarge this width. Deligne–Rapoport's normalized Tate chart is the corresponding cyclotomic coefficient ring with $[[t]]$, without inverting the level [14, VII, 1.15–1.16 and 2.1–2.4]. At primes dividing M, the same statement follows from the full-level chart by stabilizer invariants: its rotations retain exactly the powers belonging to width $3p$. Coefficient comparison in the characteristic-zero coefficient domain proves this even when the rotation order is divisible by the residue characteristic. The supersingular modifications used above leave this cusp neighborhood unchanged.

We have thus verified the all-place fraction-field embeddings of Lemma 2.2, on one branch of one cover. That lemma chooses a primitive function integral over $\mathcal{O}_L[j]$ and applies the bounded-denominator theorem to it. It proves that X_w is a congruence cover. The fine projective base group is

$$\Gamma = \pm\big(\Gamma(p) \cap \Gamma(3) \cap \Gamma_0(M)\big).$$

Put $L_0 = \mathrm{lcm}(p, 3, M)$; its p-exponent is one. Compact étaleness preserves the cusp widths, whose least common multiple divides L_0. Wohlfahrt's theorem puts $\Gamma(L_0)$ in the covering subgroup [53]. For every $g \in \mathrm{SL}_2(\mathbf{F}_p)$, the Chinese remainder theorem and surjectivity of reduction from $\mathrm{SL}_2(\mathbf{Z})$ give a lift which is g at p and the identity at the tame factors. The p-component of $\Gamma/\Gamma(L_0)$ is scalar. Conjugation by this lift is therefore the identity on that quotient. Every compact congruence p-character, and hence its Picard class w, is G-fixed. Together with (9.21) this forces $w = 0$. Thus $I_G V = 0$, and every scalar-fixed complementary point is G-invariant. The p-unit injectivity of the fine pullback gives the same conclusion on Y_2.

9.6. **Cuspidality of the marked seed.**

Proposition 9.3. *The group in* (9.4) *satisfies* $A_2 \subseteq C_2$.

Proof. The integral projector splits $Q \in A_2$ into its Steinberg and complementary parts, each scalar-fixed. First suppose $Q = eQ$. For each line L, let $f_L : Y_2 \to Y_2/B_L \simeq X_0(pM)$ be the quotient after the middle-coordinate identification. The auxiliary level has already been removed. Scalars fix every line, and on this quotient their action is σ_{a^2} on its rational $X_0(pM)$ model. Hence $f_{L,*}Q$ is F-rational. Lemma 9.2 makes it rational over $\mathbf{Q}$, and Theorem 2.1 represents it by a rational cuspidal divisor. Since $N_G Q = 0$ on the Steinberg summand, the exact identity

$$uQ = HQ = \sum_{L \in \Omega} f_L^* f_{L,*} Q$$

proves K-cuspidality of Q, using an integer inverse of the p-unit u on the finite group.

Now let Q be complementary. The preceding argument makes it G-fixed. Equation (9.3), with $\delta_a \in G$ in the full-p coordinate, then shows that Q is F-rational. Let U be an upper unipotent subgroup and let $q : Y_2 \to Y_2/U$ be its quotient. The coarse mixed-frame curve has a rational fixed cusp: take the split p-gon, put μ_p in its identity component, choose the étale generator $(1, 1)$, and put μ_M in the identity component as well. Adding a μ_p point to the étale generator is absorbed by the automorphism which multiplies the i-th component by that root of unity to the power i. Thus this cusp is fixed by U and is defined over $\mathbf{Q}$. It is a different cusp from the possibly nonrational marked analytic ∞.

The F-rational point Q is represented by an F-line bundle, since this rational cusp removes the Picard/Brauer obstruction. Over K, normalize its U-linearizations to be the identity on the fiber at that cusp. Their composites have value one there and hence satisfy the group law. Galois conjugation normalizes U and preserves the same normalization, so the linearization descends over F, even though U need only be constant over K. There is no p-inertia at an interior elliptic point: the possible elliptic orders have prime factors only $2, 3$. At every ramified cusp, cancel the fiber character with the corresponding power of the faithful tangent character. The integer exponent modulo p is Galois invariant, because both characters transform together. These corrections give an F-cuspidal divisor.

The corrected line bundle descends to Y_2/U. Its degree shows that the correction degree is divisible by p; the totally ramified rational fixed cusp adjusts both degrees integrally. We obtain

$$Q = q^*R + c, \qquad R \in \mathrm{Jac}(Y_2/U)(F), \quad c \in C_F(Y_2).$$

Pushforward and $q_* q^* = [p]$ show that R is torsion. Take primary parts to make R, c p-primary. The torus B/U has order h. Norm the displayed identity over it. Since Q is G-fixed, and if $r : Y_2/U \to Y_2/B \simeq X_0(pM)$,

$$hQ = q^* r^* (r_* R) + \sum_{b \in B/U} bc.$$

The point r_*R belongs to $J_0(pM)(F)[p^\infty]$. Lemma 9.2 and Theorem 2.1 make it a rational cuspidal class. Every term on the right is therefore K-cuspidal. Inverting h on the finite p-primary group gives $Q \in C_2$. □

9.7. **The whole higher-depth model.** We next prove the balanced relation needed for the induction.

Proposition 9.4. *For* $n \geq 3$ *and every* $Q \in A_n$,

$$(1 - S)(1 - T)Q = 0.$$

We first construct the relevant line bundle on one common integral model, including its nonreduced special fiber. At p add full level 4 over $K' = K(i)$, fix the Weil pairing, and choose a

geometrically connected component of the resulting pairing locus. The extension is unramified over K at p, and the forgetting degree divides $|\mathrm{SL}_2(\mathbf{Z}/4)| = 48$, hence is prime to p. Use the lifts

$$S_4 = \begin{pmatrix} 1 & 0 \\ -4N/p & 1 \end{pmatrix}, \qquad T_4 = \begin{pmatrix} 1 & 4/p \\ 0 & 1 \end{pmatrix}.$$

They preserve the full 4-marking, have order p, and commute. Their commutator is

$$\begin{pmatrix} 1 + 16N/p^2 & -64N/p^3 \\ -64N^2/p^3 & 1 - 16N/p^2 + 256N^2/p^4 \end{pmatrix} \in \Gamma_n \cap \Gamma(4).$$

Downstairs they induce S^4, T^4. Put

$$D_4 = (1 - S^4)(1 - T^4) = (1 - S)(1 - T)[4]_S[4]_T, \qquad [4]_S = 1 + S + S^2 + S^3. \tag{9.25}$$

The last factor has augmentation 16, a p-unit.

The unmarked fine cyclic model $\mathscr{X}$ at p is regular by [25, 6.6.1–6.6.2]; its compactification uses the normalized Tate charts. The map forgetting the two endpoints of the cyclic chain is finite on the elliptic locus. At a cusp every intermediate period is a positive rational power of the endpoint parameter times a root of unity, so its inverse j-invariant is integral in the Tate splitting extension and descends to the normal local ring. On every special component the dense ordinary locus maps by a finite Frobenius power to the ordinary locus of the middle-chain model. Thus no component is contracted. The proper extension is quasi-finite and therefore finite. Keep the transported full-4 marking on this lower model, denoted $\mathscr{Z}$. Its normalization in the fine marked function field over K' is a common model $\mathscr{Y}$. Since S_4, T_4 fix $\mathscr{Z}$, they extend to this same normalization.

We need Cartier extension of p-primary generic classes on $\mathscr{Y}$. The local calculation is as follows. At a fine supersingular point, the original completed local ring A is regular of dimension two. Katz–Mazur identify its special fiber itself as

$$k[[x, y]] \Big/ \left((x - y^{p^n})(x^{p^n} - y) \prod_{\substack{a+b=n \\ a,b \geq 1}} (x^{p^{a-1}} - y^{p^{b-1}})^{p-1} \right) \tag{9.26}$$

[25, 13.4.7–13.4.8]. The lift has the form $W(k)[[x, y]]/(f_0 + pf_1)$; regularity forces f_1 to be a unit [18, §1.1.5]. Equivalently $p \in \mathfrak{m}_A^2$ and the two lifts x, y generate the maximal ideal by Nakayama. The outer prime factors have independent linear terms $x, -y$. Their prime lifts u, v are therefore a parameter pair in the regular UFD A. The interior multiplicities are $(p-1)p^{\min(a,b)-1}$. Factoring p in this UFD and absorbing a unit into u gives

$$p = uvG^{p-1}. \tag{9.27}$$

Put $e = p - 1$ and choose the tame cyclotomic uniformizer π after an unramified coefficient extension, absorbing the unit in π^e/p. The normalization after this base change is

$$B = A[w]/(w^e - uv), \quad \pi = Gw, \qquad C = A[a, b]/(a^e - u,\ b^e - v). \tag{9.28}$$

The ring B is a hypersurface, hence S_2, and is regular at all height-one primes: at $w = u = 0$, for example, v is a unit and $u = w^e/v$. It is therefore normal and has the required fraction field. The local ring C is regular with parameters a, b. The action $(a, b) \mapsto (\zeta a, \zeta^{-1} b)$ gives $B = C^{\mu_e}$ and $w = ab$. Every nonidentity element has fixed locus only at the closed point. Thus this is a μ_e-torsor off that point.

The standard generator marking is étale in codimension one over B. On ordinary charts a connected order-p subgroup is étale-locally μ_p and its generators separate after adjoining ζ_p; for an étale subgroup this is immediate. On horizontal elliptic primes the fine marking is étale. There is no added cusp ramification: the $\Gamma_0(N)$ parabolic at denominator c, of width $N/\gcd(N, c^2)$, has diagonal variation divisible by p, since its p-valuation is $i + \max(n - 2i, 0) \geq 1$ at $i = v_p(c)$,

with the two endpoint valuations read directly. Normalize its pullback to C. It is étale away from the closed point; purity for the regular excellent two-dimensional C makes it étale everywhere [48, Tag 0BMB]. Strict henselianity splits this cover. Hence each local connected branch of $\mathscr{Y}$ is C/H_0 for a subgroup $H_0 \subset \mu_e$. Its local divisor class group is killed by $|H_0|$: pull back a divisor to factorial C, make it principal, and take its norm. Thus it is killed by e. Ordinary and cusp charts introduce no further p-class obstruction.

If a generic class has order p^r, choose a generic divisor representing it, using the based rational cusp. Choose b with $be = 1 \pmod{p^r}$, take the Weil closure of b times this divisor, and multiply by e. The result is Cartier at every point by the preceding class-group calculation, and its generic class is the original one. This gives an invertible sheaf on $\mathscr{Y}$ extending the generic class.

9.8. **The ordinary fiber and the balanced cocycle.** We work modulo the ideal (π). An ordinary cyclic p^n-chain consists of its connected steps followed by its étale steps. In a fixed middle chain of length $n-2 \geq 1$, either its first step is connected or its last step is étale. In the first case the allowed incoming extensions are the p-th roots of a fixed middle period and the source deck rotates these roots. In the second the outgoing extensions have the same description and the target deck rotates them. The backtracking extension is excluded by cyclicity.

Here are the coordinates which retain the thick fiber. For mixed type (a, b), $a, b \geq 1$ and $a + b = n$, the Tate quantity $Z = X^{p^a}_{\text{generator}}$ of [25, 13.6.3–13.6.6] is the target period: one first quotients by μ_{p^a} and then by the remaining étale generator. With endpoint periods x, y the cyclic relation is

$$\Phi_p(y^{p^{b-1}}/x^{p^{a-1}}) = 0.$$

Here $\Phi_p(X) = 1 + X + \cdots + X^{p-1}$ is the p-th cyclotomic polynomial. If $a \leq b$, set $d = b - a$, $t = y$, and $\xi = y^{p^d}/x$. Then ξ is primitive of order p^a and

$$x = \xi^{-1} t^{p^d}.$$

A good source deck sends ξ to $\zeta_p^{-c}\xi$ and fixes t; a good target sends t to $\zeta_p^c t$ and, when $d = 0$, also sends ξ to $\zeta_p^c \xi$. Both are the identity modulo (π) on the entire ring, including the nilpotents of $\mathcal{O}[\zeta_{p^a}]/(\pi)$. For $b < a$ use the symmetric coordinate. At the shallow type $(1, n-1)$ the formula is $x = \xi^{-1} t^{p^{n-2}}$, and the adjoining outer branch has $x = t^{p^n}$. The good opposite endpoint rotates t and is the identity modulo π on both branches simultaneously. The opposite shallow type is symmetric. These descriptions handle the smallest depth $n = 3$ as well.

The auxiliary prime-to-p marking is transported by the same isogenies. The generator cover over an ordinary cyclotomic chart is finite étale of degree $h = (p-1)/2 < p$. An order-p automorphism inducing the identity on the unmarked ring modulo π cannot permute its h geometric sheets nontrivially. Étale rigidity then makes it the identity on the whole marked algebra modulo π, nilpotents included.

Let L be the Cartier extension just constructed for the fine pullback of Q. Since $\mathscr{Y} \to \mathscr{Z}$ is finite, the inverse image of a sufficiently small open about each point of $\mathscr{Z}$ trivializes L: the algebra over its stalk is semilocal and a rank-one projective module there is free. These opens are invariant under both endpoint actions. On their overlaps let g_{ij} be the transition units, and form

$$d_{ij} = \frac{g_{ij}\, S_4 T_4(g_{ij})}{S_4(g_{ij}) T_4(g_{ij})}. \tag{9.29}$$

They are transitions of a line bundle L_D with generic class the pullback of $D_4 Q$. At every vertical height-one point a good action is the identity modulo π on the simultaneous chart. Commutation gives $d_{ij} \equiv 1 \pmod{\pi}$ there. At horizontal height-one points π is a unit. The functions d_{ij} and their inverses are regular; normality implies that $(d_{ij} - 1)/\pi$ is regular everywhere. Hence $L_D \bmod \pi$ is trivial across supersingular points and cusps as well.

Resolve $\mathscr{Y}$ regularly and properly, preserving the smooth based cusp, and pull back L_D. If its generic class is nonzero, let $m = p^r$ be its exact order. Then $L_D^m = \mathcal{O}(V)$ for a vertical divisor V.

Triviality modulo π gives degree zero on every component, so V has zero intersection with every component. The kernel of the intersection matrix of this connected regular fiber is generated over $\mathbf{Q}$ by the whole fiber $\mathscr{F} = \mathrm{Div}(\pi)$. The smooth cusp meets a multiplicity-one component, so $V = b\mathscr{F}$ with $b \in \mathbf{Z}$. It is principal. This proves a trivialization of L_D^m. The same multiplicity-one component gives geometric multiplicity gcd one. Raynaud's cohomological-flatness criterion implies universal $H^0(\mathcal{O}) = \mathcal{O}$ [39, 7.2.1 and 8.2.1]; in particular all global functions modulo $m\pi$ are constants.

Choose frames lifting the modulo-π trivialization, so their transitions are $1+\pi a$. For $1 \le k \le p^r$,

$$v_p\binom{p^r}{k} = r - v_p(k), \qquad (p-1)(r - v_p(k)) + k \ge (p-1)r + 1.$$

The inequality follows from $k - (p-1)v_p(k) \ge 1$. Thus the m-th powers of the transitions are 1 modulo $m\pi$. The local functions of the power trivialization glue modulo $m\pi$ to a single constant. Rigidify at the based cusp and normalize its value to one. On its true modular parameter t the power function is

$$f(t) = 1 + p^r \pi h(t), \qquad h(t) \in t\mathcal{O}[[t]]. \tag{9.30}$$

The exact valuation of the j-th binomial coefficient after this substitution is

$$v_\pi\left(\binom{1/p^r}{j}(p^r\pi)^j\right) = j - (p-1)v_p(j!) = s_p(j) \ge 1. \tag{9.31}$$

Here $s_p(j)$ is the sum of the base-p digits of j. The numerator factors $1 - kp^r$ are units, giving the first equality; Legendre's factorial formula gives the second. Since $h(0) = 0$, the series is well-defined t-adically and its based root belongs to $1 + \pi\mathcal{O}[[t]]$.

Fix globally the based Kummer cover of the class D_4Q, and pull it to the fine curve. Rigidification identifies this cover's based branch with the root just constructed. At a place away from p, extend its line bundle to a regular proper model retaining the same cusp chart. Near that smooth cusp its power divisor is a whole-fiber multiple. Dividing the local power function by its cusp value removes that factor. In the generic rigidification this cusp value is the m-th power of the value of the chosen local frame, so the normalization preserves the based cover. The result lies in $1 + t\mathcal{O}_v[[t]]$ and its unique based m-th root is integral because m is a unit there. The embeddings of the base function field into fractions of these power-series rings follow from bounded denominators of congruence forms of equal weight. Hence Lemma 2.2 applies to this one cover at all places. Its fine covering subgroup is congruence. Since fine pullback has degree prime to p, it preserves the exact order and connectedness of the p-power cover. If the fine covering subgroup $\Gamma' \cap \Gamma_{\mathrm{fine}}$ contains $\Gamma(L)$, then the coarse Γ' contains it as well. Thus the original compact cover of Y_n attached to D_4Q is congruence.

9.9. **The compact character remainder.**

Completion of the proof of Proposition 9.4. Every Γ_n cusp has the same width as its $\Gamma_0(N)$ cusp: the width parabolic already has diagonal 1 modulo p, as checked above. All widths divide N, and the width at 0 is N. For a compact étale congruence cover the least common multiple of cusp widths remains N. Wohlfahrt's theorem therefore puts $\Gamma(N)$ in the covering subgroup. Its character factors through the upper triangular group $\Gamma_n/\Gamma(N)$. The upper unipotents are generated by the parabolic $\left(\begin{smallmatrix}1 & 1\\ 0 & 1\end{smallmatrix}\right)$ at infinity; compact étaleness kills it. Thus the character factors through the full diagonal quotient, including every prime factor of M. Equivalently the cover factors through the compact unramified quotient of $X_1(N) \to Y_n$.

Let Σ_H be the Cartier dual of that constant diamond quotient. The preceding argument shows $D_4Q \in \Sigma_H(K)[p^\infty]$. Its K-rational p-primary points are killed by p, since K contains μ_p and no μ_{p^2}. Diamonds act trivially on Σ_H, because they commute with the covering diamond group; arithmetic Galois acts cyclotomically. The scalar $z_a = \delta_a\sigma_{a^2}$ therefore acts on its p-torsion by

a^2. But D_4Q is Z-fixed, since Z commutes with S, T. Choose a with $a^2 \neq 1 \pmod p$. Then $D_4Q = 0$.

Finally the element $[4]_S[4]_T$ in (9.25) is invertible on every finite p-primary module for $\langle S, T\rangle$. Modulo p its augmentation is $16 \neq 0$ and the augmentation ideal of the group algebra of the finite p-group is nilpotent, so it is a unit. Lifting to a finite group of exponent p^a gives an inverse by a finite geometric series, or by Nakayama. Apply this to the finite group generated by the S, T-orbit of Q. Equation (9.25) now gives $(1 - S)(1 - T)Q = 0$. □

9.10. **Arithmetic cusp orbits and shifted eta units.** We prove the norm statement for exactly the scalar-invariant unit group needed below. Write U_j for the K-rational functions on Y_j whose divisors are cuspidal, including $K^\times$. We first obtain an integral presentation of U_j^Z.

The arithmetic cusps for (9.2) are its orbits on primitive column vectors modulo N. In the full modular-function field the stabilizer of the based infinity cusp has first column $(\pm 1, 0)$ and unrestricted determinant; this gives the stated orbit description, with the determinant component retained. At a prime power $\ell^a \mid M$, the upper triangular GL_2 group has $a+1$ orbits, distinguished by the denominator valuation. They give $c_0 \mid M$. At p, put $i = \min(v_p(c), n)$. At $i = 0$ the upper unipotents move the numerator freely, and at $i = n$ a unit scales the numerator freely; each endpoint has one orbit. For $0 < i < n$, represent a cusp by $A/(p^i c_0)$. Its invariant is

$$\alpha = A/c_0 \pmod p \in \mathbf{F}_p^\times. \tag{9.32}$$

It is unchanged by H_Z because $a = d \pmod p$. Conversely the diagonal scalings which prescribe the two coordinates have equal reductions exactly when their ratios (9.32) agree; higher digits and the upper unipotent then complete the lift. The Chinese remainder theorem therefore gives exactly

$$c_Z(N) = \big(2 + (n-1)(p-1)\big)\tau(M) \tag{9.33}$$

arithmetic cusp orbits. These are closed cusp orbits over F, or Z-orbits of K-divisors. Their degree-zero divisor space has dimension $c_Z(N) - 1$, with the orbit degrees as weights.

For $m \mid N$ and $0 \leq k < p$ write

$$E(x) = \prod_{r \geq 1}(1 - x^r), \qquad F_{m,k}(z) = \eta(mz + k/p).$$

If $p^2 m \mid N$, then $F_{m,k}^{24} = \Delta(mz + k/p)$ has weight 12 on Γ_n. Indeed put $B_k = \left(\begin{smallmatrix} pm & k \\ 0 & p \end{smallmatrix}\right)$. For $\gamma \in \Gamma_n$, the potentially fractional upper-right entry of $B_k \gamma B_k^{-1}$ is

$$mb + (d - a)k/p - ck^2/(mp^2),$$

which is integral; the other entries are integral and the determinant is one. Its weight factor is $(cz + d)^{12}$. Thus every weight-zero product of ordinary and shifted twenty-fourth powers is a unit with trivial eta multiplier.

These weight-zero products are scalar-invariant. For a geometric diamond with diagonal (a, a^{-1}), put $k' = a^{-2}k \pmod p$. The integral conjugation $B_k \gamma_a B_{k'}^{-1}$ has upper-right entry $mb + (dk - ak')/p - ckk'/(mp^2)$. It sends the shifted Delta factor to the one indexed by k', with the same weight factor. In the modular-function arithmetic action, coefficient conjugation by a^2 sends k' back to k. Their product is the full-level scalar. Weight zero cancels the automorphy factors, proving membership in the H_Z-fixed field.

For a primitive cusp A/c the exact eta order is

$$\operatorname{ord}_{A/c} F_{m,k} = \frac{w_c \gcd(pmA + kc, pc)^2}{24p^2 m}, \qquad w_c = \frac{N}{\gcd(N, c^2)}. \tag{9.34}$$

To obtain this, multiply B_k by an integral matrix carrying infinity to A/c. Its first column has gcd $\gcd(pmA + kc, pc)$, while its determinant is $p^2 m$. The eta transformation formula gives the

displayed order, with the true cusp width w_c. For $m = p^j d$, $d \mid M$, $0 \le j \le n-2$, and nonzero k, the tame part of this gcd is $\gcd(d, c_0)$. At $c = p^i c_0$, its p-valuation is

$$\begin{cases} i, & i < j+1, \\ j+1, & i > j+1, \\ i+1, & i = j+1,\ dA + kc_0 = 0 \pmod p, \\ i, & i = j+1,\ dA + kc_0 \neq 0 \pmod p. \end{cases}$$

Consequently the divisor of $F_{p^j d,k}/F_{p^j d,k_0}$ is supported at the single height $i = j+1$, and there has coefficient

$$\frac{w_c p^{2i}(p^2-1)}{24p^2p^j}\,\frac{\gcd(c_0,d)^2}{d}\,(\mathbf{1}_{k=-d\alpha} - \mathbf{1}_{k_0=-d\alpha}). \tag{9.35}$$

Fix $k_0 = p-1$ and use $1 \le k \le p-2$. For every fixed d, the multiplication $\alpha \mapsto d\alpha$ induces an invertible change of basis on the augmentation space of functions on $\mathbf{F}_p^\times$. After these changes in the d-column blocks, (9.35) is the tensor product of that $(p-2)$-dimensional space with

$$G_{c_0,d} = \gcd(c_0,d)^2/d, \qquad c_0, d \mid M.$$

The latter matrix is invertible: the exact factorization

$$\gcd(c_0,d)^2 = \sum_{\substack{e \mid c_0 \\ e \mid d}} J_2(e), \qquad J_2(e) = e^2 \prod_{\ell \mid e}(1-\ell^{-2})$$

is an incidence-matrix times nonzero diagonal times transpose-incidence factorization. The incidence matrix, ordered by divisibility, has diagonal one. Division of the d-column by d preserves rank. This proof holds for every repeated tame factor.

Each interior height thus contributes $(p-2)\tau(M)$ independent shifted columns, and different heights have disjoint support. The ordinary weight-zero eta divisor space has rank $\tau(N)-1$, by its cusp matrix or the successive first Euler coefficients. It is constant in α. Every shifted column has sum zero in α, so these two spaces intersect trivially over $\mathbf{Q}$. Their total rank is

$$\tau(N) - 1 + (n-1)(p-2)\tau(M) = c_Z(N) - 1.$$

All columns are divisors of weight-zero products of twenty-fourth powers. They therefore span the entire invariant degree-zero cusp space over $\mathbf{Q}$.

Proposition 9.5. *Every $w \in U_j^Z$, $j \ge 2$, has a unique normalized Euler expression*

$$\begin{aligned} w(q) = cq^A &\prod_{\substack{d \mid M \\ 0 \le i \le j}} E(q^{p^i d})^{b_{i,d}} \\ &\cdot \prod_{\substack{d \mid M \\ 0 \le i \le j-2 \\ 1 \le k \le p-2}} E(\zeta_p^k q^{p^i d})^{e_{i,d,k}}, \end{aligned} \tag{9.36}$$

where $c \in K^\times$, $A \in \mathbf{Z}$, and all $b_{i,d}$, $e_{i,d,k}$ are integers. The corresponding eta exponents have total weight zero, and

$$24A = \sum_{i,d} p^i d\, b_{i,d} + \sum_{i,d,k} p^i d\, e_{i,d,k}. \tag{9.37}$$

Proof. The preceding divisor spanning gives a rational-exponent eta expression for w of total weight zero. Indeed, after clearing denominators the quotient of the corresponding unit and a power of w has zero divisor, and hence is a constant on the proper connected curve. The Euler product identity

$$\prod_{k=0}^{p-1} \eta(mz + k/p) = \text{constant}\,\frac{\eta(pmz)^{p+1}}{\eta(p^2mz)} \tag{9.38}$$

eliminates the phase $k = p-1$. To check it, for indices r prime to p take the product of $1-\zeta_p^{kr}q^{mr}$ over k, obtaining $1-q^{pmr}$; for $p \mid r$ the product is $(1-q^{mr})^p$. Collecting these terms gives the right side, and the leading q-orders agree. Since p^2m is an allowed divisor of the level, the elimination stays within the displayed support. This gives (9.36) with rational exponents, and (9.37).

Let $f = w/(cq^A)$, of leading coefficient one, and choose $D > 0$ clearing these rational exponents. The integer-exponent Euler product f^D belongs to $\mathcal{O}_K[[q]]$ and has constant coefficient one. On the other hand f has bounded denominators at every finite place: multiply w by a sufficiently large power of Δ to make it a holomorphic congruence form and use its bounded-denominator q-expansion lattice; division back by Δ preserves boundedness. This is the elementary congruence-form input used in [46, §2 and Lemma 19]. At any finite place, the Gauss valuation of a series with bounded denominators is the minimum of its coefficient valuations. It is multiplicative: scale two such series to valuation zero, reduce them to nonzero power series over the residue field, and use that their product is nonzero. Thus $Dv_{\mathrm{Gauss}}(f) = v_{\mathrm{Gauss}}(f^D) = 0$. Every coefficient of f is integral at every place, so $f \in \mathcal{O}_K[[q]]$.

For $m = p^i d$, with $p \nmid d$, write $b_m = b_{i,d}$ and $e_{m,k} = e_{i,d,k}$, taking absent entries to be zero. Order these finitely many indices increasingly. After removing smaller factors, whose exponents have already been proved integral, the coefficient of q^m is

$$-b_m - \sum_{k=1}^{p-2} e_{m,k}\zeta_p^k.$$

It is in $\mathcal{O}_K$, and $1, \zeta_p, \dots, \zeta_p^{p-2}$ is an integral $\mathbf{Z}$-basis of $\mathcal{O}_K$ and a $\mathbf{Q}$-basis of K. Its rational coordinates $b_m, e_{m,k}$ must therefore all be integers. Induction proves integrality of every exponent; applying the same first-index argument to a quotient proves uniqueness. □

9.11. **The vertical norm and all shifted congruences.**

Proposition 9.6. *For the map $\beta : Y_n \to Y_{n-1}$, $n \geq 3$,*

$$U_{n-1}^Z \cap \mathrm{Nm}_\beta K(Y_n)^\times = U_{n-1}^Z \cap \mathrm{Nm}_\beta U_n. \tag{9.39}$$

The norm on the left is taken on $K(Y_n)^\times$. Both unit groups include constants.

Proof. Complete $\mathcal{O}_K$ at $\pi = \zeta_p - 1$, obtaining $\mathcal{O}$. Let R be the completion of $\mathcal{O}[[q]]$ localized at its height-one prime (π). It is a complete DVR with residue field $\mathbf{F}_p((q))$. Put

$$L_0 = \mathrm{Frac}(R), \qquad S = R[t]/(t^p - q), \qquad L = \mathrm{Frac}(S).$$

The reduction $t^p - q$ is irreducible over $\mathbf{F}_p((q))$, so S is a DVR with uniformizer π and residue field $\mathbf{F}_p((t))$. It is the integral closure in the degree-p extension L/L_0, of ramification index one and inseparable residue degree p.

These fields compute the modular norm. A congruence modular function over K is a ratio of holomorphic congruence forms of the same weight with bounded denominators: choose a sufficiently large modular weight and a section cancelling all its poles. It therefore lies in $\mathrm{Frac}(\mathcal{O}[[q]])$. Upper functions similarly lie in $\mathrm{Frac}(\mathcal{O}[[t]])$, which embeds in L by grouping exponents modulo p. The upper function $j(t)$ is not in L_0: its coefficient of t^{p-1} in the basis $1, t, \dots, t^{p-1}$ is q^{-1} plus a nonnegative power series, and is nonzero. Since the modular extension has degree p, its base change to L_0 is exactly L. The function norm is thus this field's determinant norm.

For $z \in S$, every $\mathrm{Tr}(z^a)$ belongs to pR, since trace is p times the coefficient of 1 in its t-basis expansion. Newton's identities imply that each intermediate elementary characteristic coefficient $e_i(z)$, $1 \leq i < p$, belongs to pR: use induction in $ie_i = \sum_{a=1}^{i}(-1)^{a-1}e_{i-a}\,\mathrm{Tr}(z^a)$ and the fact that i is a unit. As $v_\pi(p) = p-1$, this gives

$$\mathrm{Nm}(1+\pi z) \equiv 1 \pmod{\pi^p}. \tag{9.40}$$

The terms $\pi^i e_i$ for $i < p$ have valuation at least $i + p - 1$, and the final term has valuation at least p. If a norm W is 1 modulo π, its witness b has valuation zero because $v_R(\operatorname{Nm} b) = pv_S(b)$. The reduction of the norm is the inseparable p-th-power map, so $\overline{b}^p = 1$ forces $\overline{b} = 1$. Thus $b = 1 + \pi z$, and (9.40) applies to every such W.

For $p \nmid d$ and every k modulo p, the exact Euler norms are

$$\operatorname{Nm} E(\zeta_p^k t^d) = E(q^d)^{p+1}/E(q^{pd}), \tag{9.41}$$

$$\operatorname{Nm} E(\zeta_p^k t^{pm}) = E(\zeta_p^k q^m)^p. \tag{9.42}$$

For the first formula, factors with index prime to p contribute $1 - q^{dr}$; indices divisible by p contribute $(1 - q^{dr/p})^p$. Collecting gives the formula. The second factor is already in the lower field, so its norm is its p-th power. These infinite products are first taken in the t-adic finite free algebra $\mathcal{O}[[t]] = \mathcal{O}[[q]][t]/(t^p - q)$. The determinant identities there are then localized and completed at (π).

Take w in the left side of (9.39), and apply Proposition 9.5 before normalizing its constant. At the unique upper cusp over the selected lower infinity, $q = t^p$. A witness with leading coefficient a has norm leading coefficient a^p: the product of the p conjugates of t^r has sign 1 for odd p. Divide the witness by a and w by a^p. We use the exponent presentation of w obtained above.

Collapse its shifted exponents into the ordinary ones, writing

$$\overline{b}_{i,d} = b_{i,d} + \sum_k e_{i,d,k}, \qquad B_d = \sum_i p^i \overline{b}_{i,d}.$$

The local function

$$v(t) = t^A \prod_{d|M} E(t^d)^{B_d}$$

has the property that $W = w/\operatorname{Nm}(v)$ is a norm and is 1 modulo π. This follows also directly by reduction: $E(q^{p^i d}) = E(q^d)^{p^i}$ in characteristic p, and (9.41) reduces to $E(q^d)$. The collapsed ordinary exponents of W have weighted sum zero on each d-axis. Consequently it has a finite product expression

$$W = \prod_m R_m(q)^{c_m} \prod_{m,k} D_{m,k}(q)^{e_{m,k}}, \quad R_m = \frac{E(q^m)^p}{E(q^{pm})}, \quad D_{m,k} = \frac{E(\zeta_p^k q^m)}{E(q^m)}. \tag{9.43}$$

All c_m are integers. More explicitly, on an axis with collapsed residual exponents $\widetilde{b}_0, \ldots, \widetilde{b}_s$ and $\sum_i p^i \widetilde{b}_i = 0$, put

$$c_i = -\sum_{j=i+1}^{s} p^{j-i-1} \widetilde{b}_j \quad (0 \le i < s).$$

Then $\widetilde{b}_0 = pc_0$, $\widetilde{b}_i = pc_i - c_{i-1}$ for $1 \le i < s$, and $\widetilde{b}_s = -c_{s-1}$, giving (9.43).

By (9.40), $W = 1 \pmod{\pi^p}$. We have $R_m = 1 \pmod p$ and $D_{m,k} = 1 \pmod \pi$; in particular $D_{m,k}^p = 1 \pmod{\pi^p}$. Induct on increasing m. Previously established p-divisible shifted exponents can be removed without changing this congruence. At q^m, reduction modulo p leaves exactly

$$-\sum_{k=1}^{p-2} e_{m,k}(\zeta_p^k - 1).$$

It must vanish. But

$$\mathcal{O}/p\mathcal{O} = \mathbf{F}_p[X]/((X-1)^{p-1}),$$

and the polynomial $\sum_{k=1}^{p-2} e_{m,k}(X^k - 1)$ has degree at most $p - 2$. Its vanishing in this quotient makes all its coefficients zero. Therefore

$$p \mid e_{i,d,k} \quad \text{for every } i, d, k. \tag{9.44}$$

The coefficientwise inference is legitimate in the vertical localization: a power series outside (π) has Gauss valuation zero, so localization and completion cannot turn a series of lower Gauss valuation into an element of $\pi^a R$.

Use (9.42) to remove each shifted factor: its norm preimage is the same phase one height higher, with exponent $e_{i,d,k}/p$. The remaining ordinary target has rational coefficients and is still a local norm. Reapply the normalization producing (9.43), now without the D factors. Its coefficients are rational and $W = 1 \pmod{\pi^p}$. Rational coefficient valuations are multiples of $p-1$, so $W = 1 \pmod{p^2}$. The first nonzero coefficient of R_m is $-p$ at q^m. Inducting on m gives $p \mid c_m$: earlier factors with p-divisible exponents are 1 modulo p^2, and the next coefficient is $-pc_m$.

In terms of the remaining ordinary target exponents, this is exactly

$$b_{0,d} + b_{1,d} = 0 \pmod{p}, \qquad b_{i,d} = 0 \pmod{p} \quad (i \geq 2). \tag{9.45}$$

In the normalization on one axis, $B = b_0 + pb_1 + \cdots + p^s b_s$, $\widetilde{b}_0 = b_0 - (p+1)B$, $\widetilde{b}_1 = b_1 + B$, and $\widetilde{b}_i = b_i$ for $i \geq 2$. The formula for c_i shows $c_i \equiv -\widetilde{b}_{i+1} \pmod{p}$. Thus all c_i vanish modulo p if and only if (9.45) holds.

The ordinary norm stencil from source exponents is

$$b_0 = (p+1)a_0 + pa_1, \qquad b_1 = -a_0 + pa_2, \qquad b_i = pa_{i+1} \quad (i \geq 2). \tag{9.46}$$

Under (9.45) an explicit integral preimage is

$$a_0 = -b_1, \qquad a_1 = \frac{b_0 + (p+1)b_1}{p}, \qquad a_2 = 0, \qquad a_{i+1} = b_i/p \quad (i \geq 2). \tag{9.47}$$

For the shifts use the preimages already specified. Every source index is allowed: a target shifted index satisfies $i \leq n-3$, so its new index $i+1 \leq n-2$ satisfies the source condition $p^2(p^{i+1}d) \mid p^n M$.

The unweighted sum of target exponents in (9.46) is p times the source sum, and their weighted sums satisfy

$$\sum_i p^i b_i = \sum_i p^i a_i.$$

Both identities hold for a shifted source term as well, by (9.42). Thus the constructed source eta product has total weight zero and infinity order A. Its twenty-fourth power is a K-unit with trivial multiplier by the matrix calculation preceding (9.34). Its leading phase is a power of ζ_p, so a K-constant normalizes this twenty-fourth power to leading coefficient one. Denote it by g_0. The exact Euler identities and infinity order give

$$\mathrm{Nm}(g_0) = w^{24}_{\text{normalized}}$$

as functions. Restoring the original constant by $g = a^{24} g_0$ gives $\mathrm{Nm}(g) = w^{24}$. Also $\mathrm{Nm}(\beta^* w) = w^p$. Choose integers r, s with $24r + ps = 1$. Then $g^r (\beta^* w)^s$ is a source cusp unit of norm w. This proves the nontrivial inclusion in (9.39); the reverse inclusion follows from pushforward of a cuspidal divisor. □

9.12. **Two descents and the full marked induction.**

Proposition 9.7. *For every $n \geq 2$, $A_n \subseteq C_n$.*

Proof. The base is Proposition 9.3. Suppose the assertion holds for $n-1$ and let $Q \in A_n$. Set $v = (T-1)Q$. Proposition 9.4 says that v is S-invariant, and it is Z-fixed because Z commutes with T.

Apply the based invariant-line-bundle descent of Section 2 to $\alpha : Y_n \to Y_{n-1}$. Its fixed K-rational totally ramified cusp normalizes the linearization. All inertia is cuspidal, so the character corrections are K-cuspidal divisors. If their initial degree is d, pullback degree shows $p \mid d$; at the fixed cusp c over b, $\alpha^* b = pc$ adjusts both degrees integrally. We obtain $v = \alpha^* R + c_0$

with a degree-zero lower class and a degree-zero cusp class. Pushforward gives pR torsion, so R is torsion. Take its p-primary part and then average this identity over Z. Invert h only on these finite p-primary groups. This makes $R \in A_{n-1}$ and retains a K-cuspidal correction. By induction $R \in C_{n-1}$, and hence

$$(T-1)Q \in C_n. \tag{9.48}$$

We now perform the different descent for Q itself. Choose K-rational degree-zero divisors A_0, E_0 representing Q and $(T-1)Q$, respectively, with E_0 cuspidal. The rational cusp supplies the divisor representative for Q. Average the *divisors*:

$$A = \sum_{z\in Z} zA_0, \qquad E = \sum_{z\in Z} zE_0, \qquad Q' = hQ.$$

They are invariant integral divisors, and there is $f \in K(Y_n)^\times$ with

$$(T-1)A - E = \mathrm{Div}(f). \tag{9.49}$$

The function can be chosen over K: a geometrically principal rational divisor has its constant Galois cocycle removed by Hilbert 90. For $z \in Z$, $z(f)/f$ is a constant in $K^\times$. These constants form a cocycle for the faithful constant-field action $\mathrm{Gal}(K/F)$. Hilbert 90 for this constant extension rescales f to be exactly Z-fixed.

Let $w = \mathrm{Nm}_\beta(f)$. Pushing forward (9.49) gives $\mathrm{Div}(w) = -\beta_* E$, since $\beta_*(T-1)A = 0$. Thus w is a lower cusp unit and is Z-fixed. Proposition 9.6 supplies $g \in U_n$ with $\mathrm{Nm}_\beta(g) = w$. Apply the cyclic Hilbert 90 formula of Section 2 to the function-field extension $K(Y_n)/K(Y_{n-1})$. It gives $t \in K(Y_n)^\times$ with $f/g = T(t)/t$. Consequently

$$(T-1)(A - \mathrm{Div}(t)) = E + \mathrm{Div}(g)$$

is supported on cusps. The noncuspidal part of $A - \mathrm{Div}(t)$ is an invariant integral divisor and descends integrally under β, which is unramified there. At an inert closed point the ramification index is one, so the integral divisor coefficient descends unchanged. Thus $A - \mathrm{Div}(t) = \beta^* D + E'$ with E' cuspidal.

At the selected rational infinity cusp, $\beta^* b = pc$. Replace the two divisors by

$$D_0 = D - (\deg D)b, \qquad E_0' = E' + p(\deg D)c.$$

Both now have degree zero, and $Q' = \beta^*[D_0] + [E_0']$. Pushforward shows $p[D_0]$ is torsion, since E_0' is cuspidal; hence $[D_0]$ is torsion. Take primary parts and average this identity over Z, inverting h on the resulting finite p-groups. The lower class is then in A_{n-1} and is cuspidal by induction. It follows that $Q' = hQ$ is in C_n, and an integer inverse of h modulo the order of Q gives $Q \in C_n$. □

Proof of Theorem 9.1. For $n = 0, 1$ use Theorem 2.1. For $n \geq 2$ let $P \in J_0(p^n M)(\mathbf{Q})[p^\infty]$. Its pullback $Q = \pi_n^* P$ belongs to A_n. By Proposition 9.7, choose a degree-zero K-cuspidal divisor D representing Q. Push it down along the degree-h marking map and sum over $\mathrm{Gal}(K/\mathbf{Q})$. This is a rational degree-zero cuspidal divisor, and its class is

$$\sum_{\sigma\in \mathrm{Gal}(K/\mathbf{Q})} \sigma\, \pi_{n,*}\pi_n^* P = h(p-1)P.$$

The multiplier is prime to p. Its integer inverse modulo the order of P expresses P itself by a rational cusp divisor. Manin–Drinfeld supplies the opposite containment. □

10. Assembly at every level

We prove Theorem 1.1. If $X_0(N)$ has genus zero, its Jacobian is zero and the assertion is immediate. In particular this covers $N = 1$.

Fix a prime p and write $N = p^n M$ with $(p, M) = 1$. For $p \geq 5$, Theorem 2.1 gives the assertion for $n = 0, 1$. The marked seed and the marked higher-depth induction of Theorem 9.1 give it

for $n \geq 2$. For $p = 3$, Theorem 8.1 covers depths 0 and 1, the $9M$ base, and the higher-depth induction over $\mathbf{Q}(\zeta_3)$. Its last trace is a trace of cuspidal divisors, and yields the rational-divisor assertion.

For $p = 2$, the cases $n = 0, 1, 2$, and $n \geq 3$ follow respectively from Theorems 3.1, 5.1, 6.1, and 7.1. The last is an induction beginning at $n = 3$, whose lower base is the $4M$ theorem. Its norm calculation treats depths 3 and 4 separately. For $M = 1$, the odd-level Jacobian is zero; the exceptional-index formula is used only for $M > 1$.

Thus for every prime p and every N we have

$$J_N(\mathbf{Q})[p^\infty] = C_{\mathrm{rat}}(N)[p^\infty].$$

Let $P \in J_N(\mathbf{Q})_{\mathrm{tors}}$. Its finite order has only finitely many prime factors, so P is the sum of its finitely many primary components. Each is represented by a degree-zero rational cuspidal divisor by the equalities just proved. The sum of these divisors represents P, proving $J_N(\mathbf{Q})_{\mathrm{tors}} \subseteq C_{\mathrm{rat}}(N)$. The two inclusions in (2.1) prove both remaining equalities. □

Appendix A. Exact computational certificates

This appendix contains all inputs, six complete executable programs, and their complete JSON outputs. Each program defines its input data and performs its calculations from those data. The matrices, integer preimages, group-ring coefficients, and exponent vectors are listed in full. JSON whitespace is compacted for typesetting; parsing the displayed text recovers the original output object exactly.

Arithmetic is performed over integers, rational numbers, explicit finite fields, and symbolic polynomial rings. The preceding sections give the general identities, geometric comparisons, and inductions to which these calculations apply.

The programs use exact Python and SymPy arithmetic. A failed assertion stops the corresponding program. Each program writes the complete output printed below to a JSON file with the same stem. The software versions used for these outputs are recorded here.

Integer matrices. For an integer matrix A, its column lattice is $A\mathbf{Z}^n$, where n is its number of columns. Swapping two rows or columns, changing one sign, and adding an integer multiple of one row or column to another are unimodular operations: their inverse is of the same form. Column operations preserve the column lattice; row operations apply an automorphism of the ambient integer lattice.

Here is the Euclidean step used in the normal forms. For two entries a, b not both zero, put $g = \gcd(a, b) > 0$ and choose integers x, y with $xa + yb = g$. The two-column replacement

$$(c_1, c_2) \longmapsto \big(xc_1 + yc_2, -(b/g)c_1 + (a/g)c_2\big)$$

has determinant $(xa + yb)/g = 1$ and changes the entries (a, b) in that row to $(g, 0)$. The integers x, y are obtained by repeated division with remainder, followed by substitution backwards.

The Smith form is the diagonal matrix obtained by such row and column operations, with positive nonzero entries $d_1 \mid d_2 \mid \cdots$. To compute it, move a nonzero entry to the first pivot position and apply Euclidean steps until the other entries in its row and column vanish. If the pivot does not divide an entry in the remaining block, add that entry's row to the pivot row and apply a column Euclidean step. The positive pivot then strictly decreases. This process terminates with a pivot dividing the remaining block. Repeat on that block. All subsequent pivots remain multiples of the preceding one, which gives the stated divisibility chain and the invariant factors.

For column Hermite normal form, work from the bottom row upwards. Use column Euclidean steps on the unprocessed columns to place the positive gcd in their rightmost column and zero in the others. Reduce the entries in this row of the already processed columns modulo that pivot by subtracting multiples of the new pivot column. That column is zero in lower rows, so their

reductions are preserved. Skip a row whose unprocessed entries are all zero; discard the zero columns at the end. These operations produce the column echelon form with positive pivots and residues between zero and the corresponding pivot minus one. Successive lowest nonzero rows determine the pivot ideals and these residues, so this form is determined by the column lattice.

The congruence-intersection routine also records its construction. Let B be a lattice basis matrix with n columns, v an integer row vector of the appropriate length, and m a positive integer. For the new condition $vBz \equiv 0 \pmod m$, with $z \in \mathbf{Z}^n$, apply the displayed column step until $vB = (g, 0, \dots, 0)$. The condition becomes $m \mid gz_1$, or $m/\gcd(m, g) \mid z_1$. Multiplying the first basis column by $m/\gcd(m, g)$ therefore gives exactly the intersection. For $g = 0$ this factor is one. Intersecting the successive conditions gives the complete congruence lattice.

Polynomials and determinants. Polynomial equalities are checked coefficient by coefficient after expansion. To reduce $f(X)$ by a monic polynomial $h(X)$, start with $r = f$. While $\deg r \geq \deg h$, subtract the leading coefficient of r times $X^{\deg r - \deg h} h(X)$ from r. This gives the unique representative of degree less than $\deg h$ in the stated quotient ring. For a monic polynomial $f(T)$ of degree d, the resultant $\mathrm{Res}_T(f, g)$ is the determinant of multiplication by g on the free algebra with basis $1, T, \dots, T^{d-1}$ modulo f. Its columns are the remainders of gT^j, $0 \leq j < d$; this is the norm used in the atomic Euler-factor calculations.

All numerical determinants can be evaluated over $\mathbf{Q}$ by the following recursion. Move a nonzero first-column entry to the upper-left corner, recording the row-swap sign, and write the matrix in blocks as $\left(\begin{smallmatrix} a & r \\ c & B \end{smallmatrix}\right)$. Subtracting c_i/a times the first row from row $i + 1$ gives

$$\det \begin{pmatrix} a & r \\ c & B \end{pmatrix} = a \det(B - cr/a).$$

A zero first column gives determinant zero, and the empty determinant is one. This recursion uses exact rational arithmetic and verifies the printed integer determinants and their decimal digit blocks.

Python: `3.12.14`; SymPy: `1.14.0`.

A.1. **Odd-level residue matrices.** Twenty-four exact integer residue matrices, their Smith diagonals, the complete cusp weights, and every weighted column sum. The listed two-exponents describe the indicated residue lattice. The general identities are (3.9) and (3.10).

Program.

```
#!/usr/bin/env python3
"""Exact checks of the integer residue matrices."""
from itertools import product
import json
from pathlib import Path
import sympy as s
from sympy.matrices.normalforms import smith_normal_form
from sympy.polys.domains import ZZ


def v2(n):
    n = abs(int(n))
    if not n:
        raise ValueError("zero has no finite valuation")
    k = 0
    while n % 2 == 0:
        n //= 2
        k += 1
    return k


def run():
    records = []
    levels = [1, 3, 5, 7, 9, 11, 15, 17, 21, 25, 27, 33, 45, 49,
              63, 65, 81, 105, 153, 175, 225, 315, 525, 945]
    for level in levels:
        ds = s.divisors(level)
        a = s.Matrix([[level * s.gcd(d, e)**2 //
                       (s.gcd(d, level // d) * d * e)
                       for e in ds] for d in ds])
        snf = smith_normal_form(a, domain=ZZ)
        actual = sorted(max(0, v2(snf[i, i]) - 3)
                        for i in range(len(ds)))
        actual = [exponent for exponent in actual if exponent]
        factors = s.factorint(level)
        expected = []
        choices = [[0] + [v2(q*q - 1)] * e for q, e in factors.items()]
        for choice in product(*choices):
            exponent = max(0, sum(choice) - 3)
            if exponent:
                expected.append(exponent)
        expected.sort()
        assert actual == expected, (level, actual, expected)
        weights = s.Matrix([[s.totient(s.gcd(d, level // d)) for d in ds]])
        psi = s.prod(q**(e - 1) * (q + 1) for q, e in factors.items())
        assert weights * a == s.ones(1, len(ds)) * psi
        records.append({"M": level, "divisors": [int(d) for d in ds],
                        "residue_matrix_times_24": [[int(x) for x in row] for row in a.tolist
  ()],
                        "smith_diagonal": [int(snf[i,i]) for i in range(len(ds))],
                        "cusp_orbit_weights": [int(x) for x in weights],
                        "weighted_column_sums": [int(x) for x in weights*a],
                        "expected_column_sum": int(psi),
```

```
                        "two_exponents_after_dividing_by_8": actual,
                        "predicted_two_exponents": expected})
    out = Path(__file__).with_suffix(".json")
    out.write_text(json.dumps(records, indent=2) + "\n")
    print(json.dumps({"checks": len(records), "status": "all exact assertions passed"}))


if __name__ == "__main__":
    run()
```

Output.


```
[
  {
    "M": 1,
    "divisors": [1],
    "residue_matrix_times_24": [[1]],
    "smith_diagonal": [1],
    "cusp_orbit_weights": [1],
    "weighted_column_sums": [1],
    "expected_column_sum": 1,
    "two_exponents_after_dividing_by_8": [],
    "predicted_two_exponents": []
  },
  {
    "M": 3,
    "divisors": [1, 3],
    "residue_matrix_times_24": [[3, 1], [1, 3]],
    "smith_diagonal": [1, 8],
    "cusp_orbit_weights": [1, 1],
    "weighted_column_sums": [4, 4],
    "expected_column_sum": 4,
    "two_exponents_after_dividing_by_8": [],
    "predicted_two_exponents": []
  },
  {
    "M": 5,
    "divisors": [1, 5],
    "residue_matrix_times_24": [[5, 1], [1, 5]],
    "smith_diagonal": [1, 24],
    "cusp_orbit_weights": [1, 1],
    "weighted_column_sums": [6, 6],
    "expected_column_sum": 6,
    "two_exponents_after_dividing_by_8": [],
    "predicted_two_exponents": []
  },
  {
    "M": 7,
    "divisors": [1, 7],
    "residue_matrix_times_24": [[7, 1], [1, 7]],
    "smith_diagonal": [1, 48],
    "cusp_orbit_weights": [1, 1],
    "weighted_column_sums": [8, 8],
    "expected_column_sum": 8,
    "two_exponents_after_dividing_by_8": [1],
    "predicted_two_exponents": [1]
  },
  {
    "M": 9,
```

```
  "divisors": [1, 3, 9],
  "residue_matrix_times_24": [[9, 3, 1], [1, 3, 1], [1, 3, 9]],
  "smith_diagonal": [1, 8, 24],
  "cusp_orbit_weights": [1, 2, 1],
  "weighted_column_sums": [12, 12, 12],
  "expected_column_sum": 12,
  "two_exponents_after_dividing_by_8": [],
  "predicted_two_exponents": []
},
{
  "M": 11,
  "divisors": [1, 11],
  "residue_matrix_times_24": [[11, 1], [1, 11]],
  "smith_diagonal": [1, 120],
  "cusp_orbit_weights": [1, 1],
  "weighted_column_sums": [12, 12],
  "expected_column_sum": 12,
  "two_exponents_after_dividing_by_8": [],
  "predicted_two_exponents": []
},
{
  "M": 15,
  "divisors": [1, 3, 5, 15],
  "residue_matrix_times_24": [[15, 5, 3, 1], [5, 15, 1, 3], [3, 1, 15, 5], [1, 3, 5, 15]],
  "smith_diagonal": [1, 8, 24, 192],
  "cusp_orbit_weights": [1, 1, 1, 1],
  "weighted_column_sums": [24, 24, 24, 24],
  "expected_column_sum": 24,
  "two_exponents_after_dividing_by_8": [3],
  "predicted_two_exponents": [3]
},
{
  "M": 17,
  "divisors": [1, 17],
  "residue_matrix_times_24": [[17, 1], [1, 17]],
  "smith_diagonal": [1, 288],
  "cusp_orbit_weights": [1, 1],
  "weighted_column_sums": [18, 18],
  "expected_column_sum": 18,
  "two_exponents_after_dividing_by_8": [2],
  "predicted_two_exponents": [2]
},
{
  "M": 21,
  "divisors": [1, 3, 7, 21],
  "residue_matrix_times_24": [[21, 7, 3, 1], [7, 21, 1, 3], [3, 1, 21, 7], [1, 3, 7, 21]],
  "smith_diagonal": [1, 8, 48, 384],
  "cusp_orbit_weights": [1, 1, 1, 1],
  "weighted_column_sums": [32, 32, 32, 32],
  "expected_column_sum": 32,
  "two_exponents_after_dividing_by_8": [1, 4],
  "predicted_two_exponents": [1, 4]
},
{
  "M": 25,
  "divisors": [1, 5, 25],
  "residue_matrix_times_24": [[25, 5, 1], [1, 5, 1], [1, 5, 25]],
  "smith_diagonal": [1, 24, 120],
  "cusp_orbit_weights": [1, 4, 1],
```

```
  "weighted_column_sums": [30, 30, 30],
  "expected_column_sum": 30,
  "two_exponents_after_dividing_by_8": [],
  "predicted_two_exponents": []
},
{
  "M": 27,
  "divisors": [1, 3, 9, 27],
  "residue_matrix_times_24": [[27, 9, 3, 1], [3, 9, 3, 1], [1, 3, 9, 3], [1, 3, 9, 27]],
  "smith_diagonal": [1, 8, 72, 72],
  "cusp_orbit_weights": [1, 2, 2, 1],
  "weighted_column_sums": [36, 36, 36, 36],
  "expected_column_sum": 36,
  "two_exponents_after_dividing_by_8": [],
  "predicted_two_exponents": []
},
{
  "M": 33,
  "divisors": [1, 3, 11, 33],
  "residue_matrix_times_24": [[33, 11, 3, 1], [11, 33, 1, 3], [3, 1, 33, 11], [1, 3, 11,
33]],
  "smith_diagonal": [1, 8, 120, 960],
  "cusp_orbit_weights": [1, 1, 1, 1],
  "weighted_column_sums": [48, 48, 48, 48],
  "expected_column_sum": 48,
  "two_exponents_after_dividing_by_8": [3],
  "predicted_two_exponents": [3]
},
{
  "M": 45,
  "divisors": [1, 3, 5, 9, 15, 45],
  "residue_matrix_times_24": [
    [45, 15, 9, 5, 3, 1],
    [5, 15, 1, 5, 3, 1],
    [9, 3, 45, 1, 15, 5],
    [5, 15, 1, 45, 3, 9],
    [1, 3, 5, 1, 15, 5],
    [1, 3, 5, 9, 15, 45]
  ],
  "smith_diagonal": [1, 8, 24, 24, 192, 576],
  "cusp_orbit_weights": [1, 2, 1, 1, 2, 1],
  "weighted_column_sums": [72, 72, 72, 72, 72, 72],
  "expected_column_sum": 72,
  "two_exponents_after_dividing_by_8": [3, 3],
  "predicted_two_exponents": [3, 3]
},
{
  "M": 49,
  "divisors": [1, 7, 49],
  "residue_matrix_times_24": [[49, 7, 1], [1, 7, 1], [1, 7, 49]],
  "smith_diagonal": [1, 48, 336],
  "cusp_orbit_weights": [1, 6, 1],
  "weighted_column_sums": [56, 56, 56],
  "expected_column_sum": 56,
  "two_exponents_after_dividing_by_8": [1, 1],
  "predicted_two_exponents": [1, 1]
},
{
  "M": 63,
```

```
  "divisors": [1, 3, 7, 9, 21, 63],
  "residue_matrix_times_24": [
    [63, 21, 9, 7, 3, 1],
    [7, 21, 1, 7, 3, 1],
    [9, 3, 63, 1, 21, 7],
    [7, 21, 1, 63, 3, 9],
    [1, 3, 7, 1, 21, 7],
    [1, 3, 7, 9, 21, 63]
  ],
  "smith_diagonal": [1, 8, 24, 48, 384, 1152],
  "cusp_orbit_weights": [1, 2, 1, 1, 2, 1],
  "weighted_column_sums": [96, 96, 96, 96, 96, 96],
  "expected_column_sum": 96,
  "two_exponents_after_dividing_by_8": [1, 4, 4],
  "predicted_two_exponents": [1, 4, 4]
},
{
  "M": 65,
  "divisors": [1, 5, 13, 65],
  "residue_matrix_times_24": [[65, 13, 5, 1], [13, 65, 1, 5], [5, 1, 65, 13], [1, 5, 13,
65]],
  "smith_diagonal": [1, 24, 168, 4032],
  "cusp_orbit_weights": [1, 1, 1, 1],
  "weighted_column_sums": [84, 84, 84, 84],
  "expected_column_sum": 84,
  "two_exponents_after_dividing_by_8": [3],
  "predicted_two_exponents": [3]
},
{
  "M": 81,
  "divisors": [1, 3, 9, 27, 81],
  "residue_matrix_times_24": [
    [81, 27, 9, 3, 1],
    [9, 27, 9, 3, 1],
    [1, 3, 9, 3, 1],
    [1, 3, 9, 27, 9],
    [1, 3, 9, 27, 81]
  ],
  "smith_diagonal": [1, 8, 72, 216, 216],
  "cusp_orbit_weights": [1, 2, 6, 2, 1],
  "weighted_column_sums": [108, 108, 108, 108, 108],
  "expected_column_sum": 108,
  "two_exponents_after_dividing_by_8": [],
  "predicted_two_exponents": []
},
{
  "M": 105,
  "divisors": [1, 3, 5, 7, 15, 21, 35, 105],
  "residue_matrix_times_24": [
    [105, 35, 21, 15, 7, 5, 3, 1],
    [35, 105, 7, 5, 21, 15, 1, 3],
    [21, 7, 105, 3, 35, 1, 15, 5],
    [15, 5, 3, 105, 1, 35, 21, 7],
    [7, 21, 35, 1, 105, 3, 5, 15],
    [5, 15, 1, 35, 3, 105, 7, 21],
    [3, 1, 15, 21, 5, 7, 105, 35],
    [1, 3, 5, 7, 15, 21, 35, 105]
  ],
  "smith_diagonal": [1, 8, 24, 48, 192, 384, 1152, 9216],
```

```
  "cusp_orbit_weights": [1, 1, 1, 1, 1, 1, 1, 1],
  "weighted_column_sums": [192, 192, 192, 192, 192, 192, 192, 192],
  "expected_column_sum": 192,
  "two_exponents_after_dividing_by_8": [1, 3, 4, 4, 7],
  "predicted_two_exponents": [1, 3, 4, 4, 7]
},
{
  "M": 153,
  "divisors": [1, 3, 9, 17, 51, 153],
  "residue_matrix_times_24": [
    [153, 51, 17, 9, 3, 1],
    [17, 51, 17, 1, 3, 1],
    [17, 51, 153, 1, 3, 9],
    [9, 3, 1, 153, 51, 17],
    [1, 3, 1, 17, 51, 17],
    [1, 3, 9, 17, 51, 153]
  ],
  "smith_diagonal": [1, 8, 24, 288, 2304, 6912],
  "cusp_orbit_weights": [1, 2, 1, 1, 2, 1],
  "weighted_column_sums": [216, 216, 216, 216, 216, 216],
  "expected_column_sum": 216,
  "two_exponents_after_dividing_by_8": [2, 5, 5],
  "predicted_two_exponents": [2, 5, 5]
},
{
  "M": 175,
  "divisors": [1, 5, 7, 25, 35, 175],
  "residue_matrix_times_24": [
    [175, 35, 25, 7, 5, 1],
    [7, 35, 1, 7, 5, 1],
    [25, 5, 175, 1, 35, 7],
    [7, 35, 1, 175, 5, 25],
    [1, 5, 7, 1, 35, 7],
    [1, 5, 7, 25, 35, 175]
  ],
  "smith_diagonal": [1, 24, 24, 48, 5760, 5760],
  "cusp_orbit_weights": [1, 4, 1, 1, 4, 1],
  "weighted_column_sums": [240, 240, 240, 240, 240, 240],
  "expected_column_sum": 240,
  "two_exponents_after_dividing_by_8": [1, 4, 4],
  "predicted_two_exponents": [1, 4, 4]
},
{
  "M": 225,
  "divisors": [1, 3, 5, 9, 15, 25, 45, 75, 225],
  "residue_matrix_times_24": [
    [225, 75, 45, 25, 15, 9, 5, 3, 1],
    [25, 75, 5, 25, 15, 1, 5, 3, 1],
    [9, 3, 45, 1, 15, 9, 5, 3, 1],
    [25, 75, 5, 225, 15, 1, 45, 3, 9],
    [1, 3, 5, 1, 15, 1, 5, 3, 1],
    [9, 3, 45, 1, 15, 225, 5, 75, 25],
    [1, 3, 5, 9, 15, 1, 45, 3, 9],
    [1, 3, 5, 1, 15, 25, 5, 75, 25],
    [1, 3, 5, 9, 15, 25, 45, 75, 225]
  ],
  "smith_diagonal": [1, 8, 24, 24, 24, 192, 960, 2880, 2880],
  "cusp_orbit_weights": [1, 2, 4, 1, 8, 1, 4, 2, 1],
  "weighted_column_sums": [360, 360, 360, 360, 360, 360, 360, 360, 360],
```

```
  "expected_column_sum": 360,
  "two_exponents_after_dividing_by_8": [3, 3, 3, 3],
  "predicted_two_exponents": [3, 3, 3, 3]
},
{
  "M": 315,
  "divisors": [1, 3, 5, 7, 9, 15, 21, 35, 45, 63, 105, 315],
  "residue_matrix_times_24": [
    [315, 105, 63, 45, 35, 21, 15, 9, 7, 5, 3, 1],
    [35, 105, 7, 5, 35, 21, 15, 1, 7, 5, 3, 1],
    [63, 21, 315, 9, 7, 105, 3, 45, 35, 1, 15, 5],
    [45, 15, 9, 315, 5, 3, 105, 63, 1, 35, 21, 7],
    [35, 105, 7, 5, 315, 21, 15, 1, 63, 45, 3, 9],
    [7, 21, 35, 1, 7, 105, 3, 5, 35, 1, 15, 5],
    [5, 15, 1, 35, 5, 3, 105, 7, 1, 35, 21, 7],
    [9, 3, 45, 63, 1, 15, 21, 315, 5, 7, 105, 35],
    [7, 21, 35, 1, 63, 105, 3, 5, 315, 9, 15, 45],
    [5, 15, 1, 35, 45, 3, 105, 7, 9, 315, 21, 63],
    [1, 3, 5, 7, 1, 15, 21, 35, 5, 7, 105, 35],
    [1, 3, 5, 7, 9, 15, 21, 35, 45, 63, 105, 315]
  ],
  "smith_diagonal": [1, 8, 24, 24, 48, 192, 192, 1152, 1152, 1152, 9216, 27648],
  "cusp_orbit_weights": [1, 2, 1, 1, 1, 2, 2, 1, 1, 1, 2, 1],
  "weighted_column_sums": [576, 576, 576, 576, 576, 576, 576, 576, 576, 576, 576, 576],
  "expected_column_sum": 576,
  "two_exponents_after_dividing_by_8": [1, 3, 3, 4, 4, 4, 7, 7],
  "predicted_two_exponents": [1, 3, 3, 4, 4, 4, 7, 7]
},
{
  "M": 525,
  "divisors": [1, 3, 5, 7, 15, 21, 25, 35, 75, 105, 175, 525],
  "residue_matrix_times_24": [
    [525, 175, 105, 75, 35, 25, 21, 15, 7, 5, 3, 1],
    [175, 525, 35, 25, 105, 75, 7, 5, 21, 15, 1, 3],
    [21, 7, 105, 3, 35, 1, 21, 15, 7, 5, 3, 1],
    [75, 25, 15, 525, 5, 175, 3, 105, 1, 35, 21, 7],
    [7, 21, 35, 1, 105, 3, 7, 5, 21, 15, 1, 3],
    [25, 75, 5, 175, 15, 525, 1, 35, 3, 105, 7, 21],
    [21, 7, 105, 3, 35, 1, 525, 15, 175, 5, 75, 25],
    [3, 1, 15, 21, 5, 7, 3, 105, 1, 35, 21, 7],
    [7, 21, 35, 1, 105, 3, 175, 5, 525, 15, 25, 75],
    [1, 3, 5, 7, 15, 21, 1, 35, 3, 105, 7, 21],
    [3, 1, 15, 21, 5, 7, 75, 105, 25, 35, 525, 175],
    [1, 3, 5, 7, 15, 21, 25, 35, 75, 105, 175, 525]
  ],
  "smith_diagonal": [1, 8, 24, 24, 48, 192, 192, 384, 5760, 5760, 46080, 46080],
  "cusp_orbit_weights": [1, 1, 4, 1, 4, 1, 1, 4, 1, 4, 1, 1],
  "weighted_column_sums": [960, 960, 960, 960, 960, 960, 960, 960, 960, 960, 960, 960],
  "expected_column_sum": 960,
  "two_exponents_after_dividing_by_8": [1, 3, 3, 4, 4, 4, 7, 7],
  "predicted_two_exponents": [1, 3, 3, 4, 4, 4, 7, 7]
},
{
  "M": 945,
  "divisors": [1, 3, 5, 7, 9, 15, 21, 27, 35, 45, 63, 105, 135, 189, 315, 945],
  "residue_matrix_times_24": [
    [945, 315, 189, 135, 105, 63, 45, 35, 27, 21, 15, 9, 7, 5, 3, 1],
    [105, 315, 21, 15, 105, 63, 45, 35, 3, 21, 15, 9, 7, 5, 3, 1],
    [189, 63, 945, 27, 21, 315, 9, 7, 135, 105, 3, 45, 35, 1, 15, 5],
```

```
      [135, 45, 27, 945, 15, 9, 315, 5, 189, 3, 105, 63, 1, 35, 21, 7],
      [35, 105, 7, 5, 315, 21, 15, 105, 1, 63, 45, 3, 21, 15, 9, 3],
      [21, 63, 105, 3, 21, 315, 9, 7, 15, 105, 3, 45, 35, 1, 15, 5],
      [15, 45, 3, 105, 15, 9, 315, 5, 21, 3, 105, 63, 1, 35, 21, 7],
      [35, 105, 7, 5, 315, 21, 15, 945, 1, 63, 45, 3, 189, 135, 9, 27],
      [27, 9, 135, 189, 3, 45, 63, 1, 945, 15, 21, 315, 5, 7, 105, 35],
      [7, 21, 35, 1, 63, 105, 3, 21, 5, 315, 9, 15, 105, 3, 45, 15],
      [5, 15, 1, 35, 45, 3, 105, 15, 7, 9, 315, 21, 3, 105, 63, 21],
      [3, 9, 15, 21, 3, 45, 63, 1, 105, 15, 21, 315, 5, 7, 105, 35],
      [7, 21, 35, 1, 63, 105, 3, 189, 5, 315, 9, 15, 945, 27, 45, 135],
      [5, 15, 1, 35, 45, 3, 105, 135, 7, 9, 315, 21, 27, 945, 63, 189],
      [1, 3, 5, 7, 9, 15, 21, 3, 35, 45, 63, 105, 15, 21, 315, 105],
      [1, 3, 5, 7, 9, 15, 21, 27, 35, 45, 63, 105, 135, 189, 315, 945]
    ],
    "smith_diagonal": [1, 8, 24, 24, 24, 48, 576, 576, 576, 1152, 3456, 3456, 3456, 27648,
  82944, 82944],
    "cusp_orbit_weights": [1, 2, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 2, 1],
    "weighted_column_sums": [
      1728, 1728, 1728, 1728, 1728, 1728, 1728, 1728, 1728, 1728, 1728, 1728, 1728,
      1728, 1728, 1728
    ],
    "expected_column_sum": 1728,
    "two_exponents_after_dividing_by_8": [1, 3, 3, 3, 4, 4, 4, 4, 7, 7, 7],
    "predicted_two_exponents": [1, 3, 3, 3, 4, 4, 4, 4, 7, 7, 7]
  }
]
```

A.2. **Dyadic models and unit norms.** Universal commutator, local-curve, Vélu, group-ring, and gluing identities; all 128 residue states at each of depths three and four; all 400 cusp-phase comparisons; and every accepted source and target exponent vector in the fixed-seed unit-norm calculations. The general phase and preimage formulas are (6.10), (7.17), (7.18), and (7.30).
Program.

```
"""Exact calculations for the dyadic descent identities.

Run with Python 3 and SymPy to evaluate the polynomial identities,
residue states, cusp phases, and eta-unit norm congruences.
"""
from fractions import Fraction as F
from pathlib import Path
import json
import itertools
import math
import random
import sympy as s

out = {}
N, h, u, t, z, a = s.symbols('N h u t z a')
A = s.Matrix([[1, 0], [3*N/2, 1]])
B = s.Matrix([[1, s.Rational(3, 2)], [0, 1]])
comm = A*B*A.inv()*B.inv()
expected = s.Matrix([[1-9*N/4, 27*N/8],
                     [-27*N*N/8, 1+9*N/4+81*N*N/16]])
assert s.simplify(comm-expected) == s.zeros(2)
comm8 = comm.subs(N, 8*h)
assert all(s.Poly(x, h).domain == s.ZZ for x in comm8)
assert s.simplify(comm8[1,0]/(8*h)).is_polynomial(h)
assert all(s.Poly((comm8-s.eye(2))[i,j], h).all_coeffs()[k] % 3 == 0
           for i in range(2) for j in range(2)
           for k in range(len(s.Poly((comm8-s.eye(2))[i,j], h).all_coeffs())))
out['commutator'] = str(comm8)

# The same A3 translation, its invariant map, and the exact local curve.
two = t*u*(u+t)
x = u*(u+t)
y = 1+u*u*(u+t)
curve_residual = s.expand(y*y-x*x*x-1)
assert s.expand(curve_residual-u*u*(u+t)*(2-two)) == 0
action = {u:t, t:-u-t}
assert s.expand(two.xreplace(action)-two) == 0
third_u = u
third_t = t
for _ in range(3):
    third_u, third_t = third_u.xreplace(action), third_t.xreplace(action)
assert s.expand(third_u-u) == 0 and s.expand(third_t-t) == 0
modular_map = (u*u+u*t+t*t)/3
assert s.expand(modular_map.xreplace(action)-modular_map) == 0
assert s.cancel((x**3+two**2)/(3*x**2)-modular_map) == 0
xx, yy = s.symbols('xx yy')
vx = (xx**3+4)/xx**2
vy = yy*(xx**3-8)/xx**3
velu = s.together(vy**2-vx**3+27).as_numer_denom()[0]
assert s.rem(velu, yy**2-xx**3-1, yy) == 0
out['prym_chart_and_order_three'] = {
    'curve_residual':str(curve_residual), 'factorization':str(u*u*(u+t)*(2-two)),
```

```
    'invariant_two':str(two),'invariant_quotient':str(modular_map),
    'third_iterate':[str(third_u),str(third_t)],
    'velu_map':[str(vx),str(vy)], 'velu_curve_residual':0}

# The regular representation verifies the exact integral S3 identity;
# the only denominator is 3, which is a unit on finite 2-groups.
perms = list(itertools.permutations(range(3)))
def comp(aa,bb):
    return tuple(aa[bb[i]] for i in range(3))
def reg(aa):
    return s.Matrix(6,6,lambda i,j:int(perms[i]==comp(aa,perms[j])))
cyc=(1,2,0)
ident=(0,1,2)
trans=[aa for aa in perms if aa!=ident and comp(aa,aa)==ident]
I=s.eye(6)
E=(I+reg(cyc)+reg(comp(cyc,cyc)))/3
assert E*E==E
assert all(reg(aa)*E==E*reg(aa) for aa in trans)
assert sum(((I+reg(aa))*(I-E) for aa in trans),s.zeros(6))==3*(I-E)
out['s3_three_old_norms'] = {
    'permutations':perms,'cycle':cyc,'transpositions':trans,
    'projector':[[str(z) for z in row] for row in E.tolist()],
    'complete_residual_matrix':[[0]*6 for _ in range(6)]}

# Exact Artin--Schreier transition identity used for the whole-model cover.
lhs = (a+(1+2*a)*z)**2 + a+(1+2*a)*z
rhs = (1+2*a)**2*(z*z+z) + a*(a+1)
assert s.expand(lhs-rhs) == 0
out['etale_gluing'] = True

# Enumerate the corrections for each shallow congruence condition.
shallow = []
for depth in (3,4):
    tested = 0
    states = []
    for T in range(8):
        for W in range(0,16,2):
            for H in range(2):
                exists = any((H-Z)%2 == 0 and
                    (4*T-2**(depth-3)*W+2**(depth-2)*Z)%8 == 0
                    for Z in range(8))
                predicted = ((H-W//2+(T if depth==4 else 0))%2 == 0)
                assert exists == predicted, (depth,T,W,H)
                states.append([T,W,H,[Z for Z in range(8) if (H-Z)%2==0 and
                    (4*T-2**(depth-3)*W+2**(depth-2)*Z)%8==0],predicted])
                tested += 1
    shallow.append({'depth':depth,'all_residue_states':tested,
        'columns':['T','W','H','all_Z_solutions_mod_8','predicted_existence'],
        'states':states})
out['shallow_kernel_elimination'] = shallow

# Derive the eta phase at the rational cusp 1/2 from an SL2 matrix
# and the Dedekind sum; compare against the phase used in the text.
def saw(x):
    return F(0) if x.denominator==1 else x-(x.numerator//x.denominator)-F(1,2)
def dedekind(d,c):
    return sum((saw(F(k,c))*saw(F(d*k,c)) for k in range(1,c)),F(0))
phase_rows=[]
for delta in range(1,401):
```

```
    c = 2
    g = math.gcd(c,delta)
    AA, CC = delta//g, c//g
    DD = pow(AA,-1,CC) if CC>1 else 0
    BB = (AA*DD-1)//CC
    assert AA*DD-BB*CC == 1
    derived = F(AA+DD,24*CC)-dedekind(DD,CC)/2-F(1,8)-F(BB*g,24*delta)
    claimed = F(delta,48)+F(g*g,48*delta)-F(1,8)
    assert (derived-claimed).denominator == 1, (delta,derived,claimed)
    phase_rows.append([delta,AA,BB,CC,DD,str(derived),str(claimed),str(derived-claimed)])
out['dedekind_phase_cases'] = {'count':400,
    'columns':['delta','A','B','C','D','derived','claimed','integer_difference'],
    'complete_table':phase_rows}

# Compute norm congruences for rational eta units.  Newman
# conditions are imposed on arbitrary source exponent vectors, including
# negative exponents, and the Euler norm stencil supplies the target.
rng = random.Random(2601126)
def factors(n):
    return [int(p) for p in s.factorint(n)]
def vp(n,p):
    v=0
    while n%p==0:
        n//=p; v+=1
    return v
def valid_unit(exps, level):
    if sum(exps.values()) != 0:return False
    if sum(d*b for d,b in exps.items())%24:return False
    if sum((level//d)*b for d,b in exps.items())%24:return False
    return all(sum(vp(d,p)*b for d,b in exps.items())%2==0 for p in factors(level))
eta_rows=[]
for depth in range(2,8):
    for M in (1,3,5,9,15):
        odd_divs=[int(d) for d in s.divisors(M)]
        keys=[(i,d) for d in odd_divs for i in range(depth+1)]
        accepted=0
        accepted_cases=[]
        for trial in range(18000):
            vals=[rng.randrange(-24,25) for _ in keys[:-1]]
            vals.append(-sum(vals))
            av=dict(zip(keys,vals))
            eta={2**i*d:v for (i,d),v in av.items()}
            if not valid_unit(eta,2**depth*M):continue
            b={}
            for d in odd_divs:
                b[0,d]=3*av[0,d]+2*av[1,d]
                b[1,d]=-av[0,d]+2*av[2,d]
                for i in range(2,depth):b[i,d]=2*av[i+1,d]
            target={2**i*d:v for (i,d),v in b.items()}
            assert valid_unit(target,2**(depth-1)*M)
            assert all((b[0,d]-b[1,d])%2==0 for d in odd_divs)
            assert all(v%2==0 for (i,d),v in b.items() if i>=2)
            assert all(sum(vp(d,p)*v for (i,d),v in b.items())%4==0 for p in factors(M))
            if depth>=3:
                reciprocal=sum((2**(depth-1)*M//(2**i*d))*v for (i,d),v in b.items())
                T=reciprocal//24
                W=sum((M//d)*b[1,d] for d in odd_divs)
                H=(sum(i*v for (i,d),v in b.items())+sum(b[1,d] for d in odd_divs))//2
                assert W%2==0
```

```
                assert sum(b[depth-1,d] for d in odd_divs)%4==0
                assert (T%2==0 if depth>=5 else (H-W//2+(T if depth==4 else 0))%2==0)
            accepted+=1
            accepted_cases.append([trial+1,vals,
                [b[i,d] for d in odd_divs for i in range(depth)],
                sum(target.values()),sum(d*v for d,v in target.items()),
                sum((2**(depth-1)*M//d)*v for d,v in target.items())])
            if accepted>=64:break
        assert accepted>0, (depth,M)
        eta_rows.append({'depth':depth,'odd_part':M,'valid_source_units':accepted,
                         'trials':trial+1,'source_coordinates':keys,
                         'target_coordinates':[(i,d) for d in odd_divs for i in range(depth)],
                         'columns':['trial','source_exponents','target_exponents',
                                    'target_weight_sum','target_infinity_sum','
  target_reciprocal_sum'],
                         'accepted_cases':accepted_cases})
out['eta_norm_cases']=eta_rows
out['total_valid_source_units']=sum(row['valid_source_units'] for row in eta_rows)
out['conclusion']='All exact assertions passed.'
dest=Path(__file__).with_suffix('.json')
dest.write_text(json.dumps(out,ensure_ascii=False,indent=2)+'\n')
print(json.dumps(out,ensure_ascii=False,indent=2))
```

Output.


```
{
  "commutator": "Matrix([[1 - 18*h, 27*h], [-12*h*(18*h + 1) + 12*h, 18*h*(18*h + 1) + 1]])",
  "prym_chart_and_order_three": {
    "curve_residual": "-t**3*u**3 - 2*t**2*u**4 - t*u**5 + 2*t*u**2 + 2*u**3",
    "factorization": "u**2*(t + u)*(-t*u*(t + u) + 2)",
    "invariant_two": "t*u*(t + u)",
    "invariant_quotient": "t**2/3 + t*u/3 + u**2/3",
    "third_iterate": ["u", "t"],
    "velu_map": ["(xx**3 + 4)/xx**2", "yy*(xx**3 - 8)/xx**3"],
    "velu_curve_residual": 0
  },
  "s3_three_old_norms": {
    "permutations": [[0, 1, 2], [0, 2, 1], [1, 0, 2], [1, 2, 0], [2, 0, 1], [2, 1, 0]],
    "cycle": [1, 2, 0],
    "transpositions": [[0, 2, 1], [1, 0, 2], [2, 1, 0]],
    "projector": [
      ["1/3", "0", "0", "1/3", "1/3", "0"],
      ["0", "1/3", "1/3", "0", "0", "1/3"],
      ["0", "1/3", "1/3", "0", "0", "1/3"],
      ["1/3", "0", "0", "1/3", "1/3", "0"],
      ["1/3", "0", "0", "1/3", "1/3", "0"],
      ["0", "1/3", "1/3", "0", "0", "1/3"]
    ],
    "complete_residual_matrix": [
      [0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0]
    ]
  },
  "etale_gluing": true,
  "shallow_kernel_elimination": [
```

```
{
  "depth": 3,
  "all_residue_states": 128,
  "columns": ["T", "W", "H", "all_Z_solutions_mod_8", "predicted_existence"],
  "states": [
    [0, 0, 0, [0, 4], true],
    [0, 0, 1, [], false],
    [0, 2, 0, [], false],
    [0, 2, 1, [1, 5], true],
    [0, 4, 0, [2, 6], true],
    [0, 4, 1, [], false],
    [0, 6, 0, [], false],
    [0, 6, 1, [3, 7], true],
    [0, 8, 0, [0, 4], true],
    [0, 8, 1, [], false],
    [0, 10, 0, [], false],
    [0, 10, 1, [1, 5], true],
    [0, 12, 0, [2, 6], true],
    [0, 12, 1, [], false],
    [0, 14, 0, [], false],
    [0, 14, 1, [3, 7], true],
    [1, 0, 0, [2, 6], true],
    [1, 0, 1, [], false],
    [1, 2, 0, [], false],
    [1, 2, 1, [3, 7], true],
    [1, 4, 0, [0, 4], true],
    [1, 4, 1, [], false],
    [1, 6, 0, [], false],
    [1, 6, 1, [1, 5], true],
    [1, 8, 0, [2, 6], true],
    [1, 8, 1, [], false],
    [1, 10, 0, [], false],
    [1, 10, 1, [3, 7], true],
    [1, 12, 0, [0, 4], true],
    [1, 12, 1, [], false],
    [1, 14, 0, [], false],
    [1, 14, 1, [1, 5], true],
    [2, 0, 0, [0, 4], true],
    [2, 0, 1, [], false],
    [2, 2, 0, [], false],
    [2, 2, 1, [1, 5], true],
    [2, 4, 0, [2, 6], true],
    [2, 4, 1, [], false],
    [2, 6, 0, [], false],
    [2, 6, 1, [3, 7], true],
    [2, 8, 0, [0, 4], true],
    [2, 8, 1, [], false],
    [2, 10, 0, [], false],
    [2, 10, 1, [1, 5], true],
    [2, 12, 0, [2, 6], true],
    [2, 12, 1, [], false],
    [2, 14, 0, [], false],
    [2, 14, 1, [3, 7], true],
    [3, 0, 0, [2, 6], true],
    [3, 0, 1, [], false],
    [3, 2, 0, [], false],
    [3, 2, 1, [3, 7], true],
    [3, 4, 0, [0, 4], true],
    [3, 4, 1, [], false],
```

```
[3, 6, 0, [], false],
[3, 6, 1, [1, 5], true],
[3, 8, 0, [2, 6], true],
[3, 8, 1, [], false],
[3, 10, 0, [], false],
[3, 10, 1, [3, 7], true],
[3, 12, 0, [0, 4], true],
[3, 12, 1, [], false],
[3, 14, 0, [], false],
[3, 14, 1, [1, 5], true],
[4, 0, 0, [0, 4], true],
[4, 0, 1, [], false],
[4, 2, 0, [], false],
[4, 2, 1, [1, 5], true],
[4, 4, 0, [2, 6], true],
[4, 4, 1, [], false],
[4, 6, 0, [], false],
[4, 6, 1, [3, 7], true],
[4, 8, 0, [0, 4], true],
[4, 8, 1, [], false],
[4, 10, 0, [], false],
[4, 10, 1, [1, 5], true],
[4, 12, 0, [2, 6], true],
[4, 12, 1, [], false],
[4, 14, 0, [], false],
[4, 14, 1, [3, 7], true],
[5, 0, 0, [2, 6], true],
[5, 0, 1, [], false],
[5, 2, 0, [], false],
[5, 2, 1, [3, 7], true],
[5, 4, 0, [0, 4], true],
[5, 4, 1, [], false],
[5, 6, 0, [], false],
[5, 6, 1, [1, 5], true],
[5, 8, 0, [2, 6], true],
[5, 8, 1, [], false],
[5, 10, 0, [], false],
[5, 10, 1, [3, 7], true],
[5, 12, 0, [0, 4], true],
[5, 12, 1, [], false],
[5, 14, 0, [], false],
[5, 14, 1, [1, 5], true],
[6, 0, 0, [0, 4], true],
[6, 0, 1, [], false],
[6, 2, 0, [], false],
[6, 2, 1, [1, 5], true],
[6, 4, 0, [2, 6], true],
[6, 4, 1, [], false],
[6, 6, 0, [], false],
[6, 6, 1, [3, 7], true],
[6, 8, 0, [0, 4], true],
[6, 8, 1, [], false],
[6, 10, 0, [], false],
[6, 10, 1, [1, 5], true],
[6, 12, 0, [2, 6], true],
[6, 12, 1, [], false],
[6, 14, 0, [], false],
[6, 14, 1, [3, 7], true],
[7, 0, 0, [2, 6], true],
```

```
      [7, 0, 1, [], false],
      [7, 2, 0, [], false],
      [7, 2, 1, [3, 7], true],
      [7, 4, 0, [0, 4], true],
      [7, 4, 1, [], false],
      [7, 6, 0, [], false],
      [7, 6, 1, [1, 5], true],
      [7, 8, 0, [2, 6], true],
      [7, 8, 1, [], false],
      [7, 10, 0, [], false],
      [7, 10, 1, [3, 7], true],
      [7, 12, 0, [0, 4], true],
      [7, 12, 1, [], false],
      [7, 14, 0, [], false],
      [7, 14, 1, [1, 5], true]
    ]
  },
  {
    "depth": 4,
    "all_residue_states": 128,
    "columns": ["T", "W", "H", "all_Z_solutions_mod_8", "predicted_existence"],
    "states": [
      [0, 0, 0, [0, 2, 4, 6], true],
      [0, 0, 1, [], false],
      [0, 2, 0, [], false],
      [0, 2, 1, [1, 3, 5, 7], true],
      [0, 4, 0, [0, 2, 4, 6], true],
      [0, 4, 1, [], false],
      [0, 6, 0, [], false],
      [0, 6, 1, [1, 3, 5, 7], true],
      [0, 8, 0, [0, 2, 4, 6], true],
      [0, 8, 1, [], false],
      [0, 10, 0, [], false],
      [0, 10, 1, [1, 3, 5, 7], true],
      [0, 12, 0, [0, 2, 4, 6], true],
      [0, 12, 1, [], false],
      [0, 14, 0, [], false],
      [0, 14, 1, [1, 3, 5, 7], true],
      [1, 0, 0, [], false],
      [1, 0, 1, [1, 3, 5, 7], true],
      [1, 2, 0, [0, 2, 4, 6], true],
      [1, 2, 1, [], false],
      [1, 4, 0, [], false],
      [1, 4, 1, [1, 3, 5, 7], true],
      [1, 6, 0, [0, 2, 4, 6], true],
      [1, 6, 1, [], false],
      [1, 8, 0, [], false],
      [1, 8, 1, [1, 3, 5, 7], true],
      [1, 10, 0, [0, 2, 4, 6], true],
      [1, 10, 1, [], false],
      [1, 12, 0, [], false],
      [1, 12, 1, [1, 3, 5, 7], true],
      [1, 14, 0, [0, 2, 4, 6], true],
      [1, 14, 1, [], false],
      [2, 0, 0, [0, 2, 4, 6], true],
      [2, 0, 1, [], false],
      [2, 2, 0, [], false],
      [2, 2, 1, [1, 3, 5, 7], true],
      [2, 4, 0, [0, 2, 4, 6], true],
```

```
[2, 4, 1, [], false],
[2, 6, 0, [], false],
[2, 6, 1, [1, 3, 5, 7], true],
[2, 8, 0, [0, 2, 4, 6], true],
[2, 8, 1, [], false],
[2, 10, 0, [], false],
[2, 10, 1, [1, 3, 5, 7], true],
[2, 12, 0, [0, 2, 4, 6], true],
[2, 12, 1, [], false],
[2, 14, 0, [], false],
[2, 14, 1, [1, 3, 5, 7], true],
[3, 0, 0, [], false],
[3, 0, 1, [1, 3, 5, 7], true],
[3, 2, 0, [0, 2, 4, 6], true],
[3, 2, 1, [], false],
[3, 4, 0, [], false],
[3, 4, 1, [1, 3, 5, 7], true],
[3, 6, 0, [0, 2, 4, 6], true],
[3, 6, 1, [], false],
[3, 8, 0, [], false],
[3, 8, 1, [1, 3, 5, 7], true],
[3, 10, 0, [0, 2, 4, 6], true],
[3, 10, 1, [], false],
[3, 12, 0, [], false],
[3, 12, 1, [1, 3, 5, 7], true],
[3, 14, 0, [0, 2, 4, 6], true],
[3, 14, 1, [], false],
[4, 0, 0, [0, 2, 4, 6], true],
[4, 0, 1, [], false],
[4, 2, 0, [], false],
[4, 2, 1, [1, 3, 5, 7], true],
[4, 4, 0, [0, 2, 4, 6], true],
[4, 4, 1, [], false],
[4, 6, 0, [], false],
[4, 6, 1, [1, 3, 5, 7], true],
[4, 8, 0, [0, 2, 4, 6], true],
[4, 8, 1, [], false],
[4, 10, 0, [], false],
[4, 10, 1, [1, 3, 5, 7], true],
[4, 12, 0, [0, 2, 4, 6], true],
[4, 12, 1, [], false],
[4, 14, 0, [], false],
[4, 14, 1, [1, 3, 5, 7], true],
[5, 0, 0, [], false],
[5, 0, 1, [1, 3, 5, 7], true],
[5, 2, 0, [0, 2, 4, 6], true],
[5, 2, 1, [], false],
[5, 4, 0, [], false],
[5, 4, 1, [1, 3, 5, 7], true],
[5, 6, 0, [0, 2, 4, 6], true],
[5, 6, 1, [], false],
[5, 8, 0, [], false],
[5, 8, 1, [1, 3, 5, 7], true],
[5, 10, 0, [0, 2, 4, 6], true],
[5, 10, 1, [], false],
[5, 12, 0, [], false],
[5, 12, 1, [1, 3, 5, 7], true],
[5, 14, 0, [0, 2, 4, 6], true],
[5, 14, 1, [], false],
```

```
      [6, 0, 0, [0, 2, 4, 6], true],
      [6, 0, 1, [], false],
      [6, 2, 0, [], false],
      [6, 2, 1, [1, 3, 5, 7], true],
      [6, 4, 0, [0, 2, 4, 6], true],
      [6, 4, 1, [], false],
      [6, 6, 0, [], false],
      [6, 6, 1, [1, 3, 5, 7], true],
      [6, 8, 0, [0, 2, 4, 6], true],
      [6, 8, 1, [], false],
      [6, 10, 0, [], false],
      [6, 10, 1, [1, 3, 5, 7], true],
      [6, 12, 0, [0, 2, 4, 6], true],
      [6, 12, 1, [], false],
      [6, 14, 0, [], false],
      [6, 14, 1, [1, 3, 5, 7], true],
      [7, 0, 0, [], false],
      [7, 0, 1, [1, 3, 5, 7], true],
      [7, 2, 0, [0, 2, 4, 6], true],
      [7, 2, 1, [], false],
      [7, 4, 0, [], false],
      [7, 4, 1, [1, 3, 5, 7], true],
      [7, 6, 0, [0, 2, 4, 6], true],
      [7, 6, 1, [], false],
      [7, 8, 0, [], false],
      [7, 8, 1, [1, 3, 5, 7], true],
      [7, 10, 0, [0, 2, 4, 6], true],
      [7, 10, 1, [], false],
      [7, 12, 0, [], false],
      [7, 12, 1, [1, 3, 5, 7], true],
      [7, 14, 0, [0, 2, 4, 6], true],
      [7, 14, 1, [], false]
    ]
  }
],
"dedekind_phase_cases": {
  "count": 400,
  "columns": ["delta", "A", "B", "C", "D", "derived", "claimed", "integer_difference"],
  "complete_table": [
    [1, 1, 0, 2, 1, "-1/12", "-1/12", "0"],
    [2, 1, -1, 1, 0, "-1/24", "-1/24", "0"],
    [3, 3, 1, 2, 1, "-1/18", "-1/18", "0"],
    [4, 2, -1, 1, 0, "-1/48", "-1/48", "0"],
    [5, 5, 2, 2, 1, "-1/60", "-1/60", "0"],
    [6, 3, -1, 1, 0, "1/72", "1/72", "0"],
    [7, 7, 3, 2, 1, "1/42", "1/42", "0"],
    [8, 4, -1, 1, 0, "5/96", "5/96", "0"],
    [9, 9, 4, 2, 1, "7/108", "7/108", "0"],
    [10, 5, -1, 1, 0, "11/120", "11/120", "0"],
    [11, 11, 5, 2, 1, "7/66", "7/66", "0"],
    [12, 6, -1, 1, 0, "19/144", "19/144", "0"],
    [13, 13, 6, 2, 1, "23/156", "23/156", "0"],
    [14, 7, -1, 1, 0, "29/168", "29/168", "0"],
    [15, 15, 7, 2, 1, "17/90", "17/90", "0"],
    [16, 8, -1, 1, 0, "41/192", "41/192", "0"],
    [17, 17, 8, 2, 1, "47/204", "47/204", "0"],
    [18, 9, -1, 1, 0, "55/216", "55/216", "0"],
    [19, 19, 9, 2, 1, "31/114", "31/114", "0"],
    [20, 10, -1, 1, 0, "71/240", "71/240", "0"],
```

```
[21, 21, 10, 2, 1, "79/252", "79/252", "0"],
[22, 11, -1, 1, 0, "89/264", "89/264", "0"],
[23, 23, 11, 2, 1, "49/138", "49/138", "0"],
[24, 12, -1, 1, 0, "109/288", "109/288", "0"],
[25, 25, 12, 2, 1, "119/300", "119/300", "0"],
[26, 13, -1, 1, 0, "131/312", "131/312", "0"],
[27, 27, 13, 2, 1, "71/162", "71/162", "0"],
[28, 14, -1, 1, 0, "155/336", "155/336", "0"],
[29, 29, 14, 2, 1, "167/348", "167/348", "0"],
[30, 15, -1, 1, 0, "181/360", "181/360", "0"],
[31, 31, 15, 2, 1, "97/186", "97/186", "0"],
[32, 16, -1, 1, 0, "209/384", "209/384", "0"],
[33, 33, 16, 2, 1, "223/396", "223/396", "0"],
[34, 17, -1, 1, 0, "239/408", "239/408", "0"],
[35, 35, 17, 2, 1, "127/210", "127/210", "0"],
[36, 18, -1, 1, 0, "271/432", "271/432", "0"],
[37, 37, 18, 2, 1, "287/444", "287/444", "0"],
[38, 19, -1, 1, 0, "305/456", "305/456", "0"],
[39, 39, 19, 2, 1, "161/234", "161/234", "0"],
[40, 20, -1, 1, 0, "341/480", "341/480", "0"],
[41, 41, 20, 2, 1, "359/492", "359/492", "0"],
[42, 21, -1, 1, 0, "379/504", "379/504", "0"],
[43, 43, 21, 2, 1, "199/258", "199/258", "0"],
[44, 22, -1, 1, 0, "419/528", "419/528", "0"],
[45, 45, 22, 2, 1, "439/540", "439/540", "0"],
[46, 23, -1, 1, 0, "461/552", "461/552", "0"],
[47, 47, 23, 2, 1, "241/282", "241/282", "0"],
[48, 24, -1, 1, 0, "505/576", "505/576", "0"],
[49, 49, 24, 2, 1, "527/588", "527/588", "0"],
[50, 25, -1, 1, 0, "551/600", "551/600", "0"],
[51, 51, 25, 2, 1, "287/306", "287/306", "0"],
[52, 26, -1, 1, 0, "599/624", "599/624", "0"],
[53, 53, 26, 2, 1, "623/636", "623/636", "0"],
[54, 27, -1, 1, 0, "649/648", "649/648", "0"],
[55, 55, 27, 2, 1, "337/330", "337/330", "0"],
[56, 28, -1, 1, 0, "701/672", "701/672", "0"],
[57, 57, 28, 2, 1, "727/684", "727/684", "0"],
[58, 29, -1, 1, 0, "755/696", "755/696", "0"],
[59, 59, 29, 2, 1, "391/354", "391/354", "0"],
[60, 30, -1, 1, 0, "811/720", "811/720", "0"],
[61, 61, 30, 2, 1, "839/732", "839/732", "0"],
[62, 31, -1, 1, 0, "869/744", "869/744", "0"],
[63, 63, 31, 2, 1, "449/378", "449/378", "0"],
[64, 32, -1, 1, 0, "929/768", "929/768", "0"],
[65, 65, 32, 2, 1, "959/780", "959/780", "0"],
[66, 33, -1, 1, 0, "991/792", "991/792", "0"],
[67, 67, 33, 2, 1, "511/402", "511/402", "0"],
[68, 34, -1, 1, 0, "1055/816", "1055/816", "0"],
[69, 69, 34, 2, 1, "1087/828", "1087/828", "0"],
[70, 35, -1, 1, 0, "1121/840", "1121/840", "0"],
[71, 71, 35, 2, 1, "577/426", "577/426", "0"],
[72, 36, -1, 1, 0, "1189/864", "1189/864", "0"],
[73, 73, 36, 2, 1, "1223/876", "1223/876", "0"],
[74, 37, -1, 1, 0, "1259/888", "1259/888", "0"],
[75, 75, 37, 2, 1, "647/450", "647/450", "0"],
[76, 38, -1, 1, 0, "1331/912", "1331/912", "0"],
[77, 77, 38, 2, 1, "1367/924", "1367/924", "0"],
[78, 39, -1, 1, 0, "1405/936", "1405/936", "0"],
[79, 79, 39, 2, 1, "721/474", "721/474", "0"],
```

```
[80, 40, -1, 1, 0, "1481/960", "1481/960", "0"],
[81, 81, 40, 2, 1, "1519/972", "1519/972", "0"],
[82, 41, -1, 1, 0, "1559/984", "1559/984", "0"],
[83, 83, 41, 2, 1, "799/498", "799/498", "0"],
[84, 42, -1, 1, 0, "1639/1008", "1639/1008", "0"],
[85, 85, 42, 2, 1, "1679/1020", "1679/1020", "0"],
[86, 43, -1, 1, 0, "1721/1032", "1721/1032", "0"],
[87, 87, 43, 2, 1, "881/522", "881/522", "0"],
[88, 44, -1, 1, 0, "1805/1056", "1805/1056", "0"],
[89, 89, 44, 2, 1, "1847/1068", "1847/1068", "0"],
[90, 45, -1, 1, 0, "1891/1080", "1891/1080", "0"],
[91, 91, 45, 2, 1, "967/546", "967/546", "0"],
[92, 46, -1, 1, 0, "1979/1104", "1979/1104", "0"],
[93, 93, 46, 2, 1, "2023/1116", "2023/1116", "0"],
[94, 47, -1, 1, 0, "2069/1128", "2069/1128", "0"],
[95, 95, 47, 2, 1, "1057/570", "1057/570", "0"],
[96, 48, -1, 1, 0, "2161/1152", "2161/1152", "0"],
[97, 97, 48, 2, 1, "2207/1164", "2207/1164", "0"],
[98, 49, -1, 1, 0, "2255/1176", "2255/1176", "0"],
[99, 99, 49, 2, 1, "1151/594", "1151/594", "0"],
[100, 50, -1, 1, 0, "2351/1200", "2351/1200", "0"],
[101, 101, 50, 2, 1, "2399/1212", "2399/1212", "0"],
[102, 51, -1, 1, 0, "2449/1224", "2449/1224", "0"],
[103, 103, 51, 2, 1, "1249/618", "1249/618", "0"],
[104, 52, -1, 1, 0, "2549/1248", "2549/1248", "0"],
[105, 105, 52, 2, 1, "2599/1260", "2599/1260", "0"],
[106, 53, -1, 1, 0, "2651/1272", "2651/1272", "0"],
[107, 107, 53, 2, 1, "1351/642", "1351/642", "0"],
[108, 54, -1, 1, 0, "2755/1296", "2755/1296", "0"],
[109, 109, 54, 2, 1, "2807/1308", "2807/1308", "0"],
[110, 55, -1, 1, 0, "2861/1320", "2861/1320", "0"],
[111, 111, 55, 2, 1, "1457/666", "1457/666", "0"],
[112, 56, -1, 1, 0, "2969/1344", "2969/1344", "0"],
[113, 113, 56, 2, 1, "3023/1356", "3023/1356", "0"],
[114, 57, -1, 1, 0, "3079/1368", "3079/1368", "0"],
[115, 115, 57, 2, 1, "1567/690", "1567/690", "0"],
[116, 58, -1, 1, 0, "3191/1392", "3191/1392", "0"],
[117, 117, 58, 2, 1, "3247/1404", "3247/1404", "0"],
[118, 59, -1, 1, 0, "3305/1416", "3305/1416", "0"],
[119, 119, 59, 2, 1, "1681/714", "1681/714", "0"],
[120, 60, -1, 1, 0, "3421/1440", "3421/1440", "0"],
[121, 121, 60, 2, 1, "3479/1452", "3479/1452", "0"],
[122, 61, -1, 1, 0, "3539/1464", "3539/1464", "0"],
[123, 123, 61, 2, 1, "1799/738", "1799/738", "0"],
[124, 62, -1, 1, 0, "3659/1488", "3659/1488", "0"],
[125, 125, 62, 2, 1, "3719/1500", "3719/1500", "0"],
[126, 63, -1, 1, 0, "3781/1512", "3781/1512", "0"],
[127, 127, 63, 2, 1, "1921/762", "1921/762", "0"],
[128, 64, -1, 1, 0, "3905/1536", "3905/1536", "0"],
[129, 129, 64, 2, 1, "3967/1548", "3967/1548", "0"],
[130, 65, -1, 1, 0, "4031/1560", "4031/1560", "0"],
[131, 131, 65, 2, 1, "2047/786", "2047/786", "0"],
[132, 66, -1, 1, 0, "4159/1584", "4159/1584", "0"],
[133, 133, 66, 2, 1, "4223/1596", "4223/1596", "0"],
[134, 67, -1, 1, 0, "4289/1608", "4289/1608", "0"],
[135, 135, 67, 2, 1, "2177/810", "2177/810", "0"],
[136, 68, -1, 1, 0, "4421/1632", "4421/1632", "0"],
[137, 137, 68, 2, 1, "4487/1644", "4487/1644", "0"],
[138, 69, -1, 1, 0, "4555/1656", "4555/1656", "0"],
```

```
[139, 139, 69, 2, 1, "2311/834", "2311/834", "0"],
[140, 70, -1, 1, 0, "4691/1680", "4691/1680", "0"],
[141, 141, 70, 2, 1, "4759/1692", "4759/1692", "0"],
[142, 71, -1, 1, 0, "4829/1704", "4829/1704", "0"],
[143, 143, 71, 2, 1, "2449/858", "2449/858", "0"],
[144, 72, -1, 1, 0, "4969/1728", "4969/1728", "0"],
[145, 145, 72, 2, 1, "5039/1740", "5039/1740", "0"],
[146, 73, -1, 1, 0, "5111/1752", "5111/1752", "0"],
[147, 147, 73, 2, 1, "2591/882", "2591/882", "0"],
[148, 74, -1, 1, 0, "5255/1776", "5255/1776", "0"],
[149, 149, 74, 2, 1, "5327/1788", "5327/1788", "0"],
[150, 75, -1, 1, 0, "5401/1800", "5401/1800", "0"],
[151, 151, 75, 2, 1, "2737/906", "2737/906", "0"],
[152, 76, -1, 1, 0, "5549/1824", "5549/1824", "0"],
[153, 153, 76, 2, 1, "5623/1836", "5623/1836", "0"],
[154, 77, -1, 1, 0, "5699/1848", "5699/1848", "0"],
[155, 155, 77, 2, 1, "2887/930", "2887/930", "0"],
[156, 78, -1, 1, 0, "5851/1872", "5851/1872", "0"],
[157, 157, 78, 2, 1, "5927/1884", "5927/1884", "0"],
[158, 79, -1, 1, 0, "6005/1896", "6005/1896", "0"],
[159, 159, 79, 2, 1, "3041/954", "3041/954", "0"],
[160, 80, -1, 1, 0, "6161/1920", "6161/1920", "0"],
[161, 161, 80, 2, 1, "6239/1932", "6239/1932", "0"],
[162, 81, -1, 1, 0, "6319/1944", "6319/1944", "0"],
[163, 163, 81, 2, 1, "3199/978", "3199/978", "0"],
[164, 82, -1, 1, 0, "6479/1968", "6479/1968", "0"],
[165, 165, 82, 2, 1, "6559/1980", "6559/1980", "0"],
[166, 83, -1, 1, 0, "6641/1992", "6641/1992", "0"],
[167, 167, 83, 2, 1, "3361/1002", "3361/1002", "0"],
[168, 84, -1, 1, 0, "6805/2016", "6805/2016", "0"],
[169, 169, 84, 2, 1, "6887/2028", "6887/2028", "0"],
[170, 85, -1, 1, 0, "6971/2040", "6971/2040", "0"],
[171, 171, 85, 2, 1, "3527/1026", "3527/1026", "0"],
[172, 86, -1, 1, 0, "7139/2064", "7139/2064", "0"],
[173, 173, 86, 2, 1, "7223/2076", "7223/2076", "0"],
[174, 87, -1, 1, 0, "7309/2088", "7309/2088", "0"],
[175, 175, 87, 2, 1, "3697/1050", "3697/1050", "0"],
[176, 88, -1, 1, 0, "7481/2112", "7481/2112", "0"],
[177, 177, 88, 2, 1, "7567/2124", "7567/2124", "0"],
[178, 89, -1, 1, 0, "7655/2136", "7655/2136", "0"],
[179, 179, 89, 2, 1, "3871/1074", "3871/1074", "0"],
[180, 90, -1, 1, 0, "7831/2160", "7831/2160", "0"],
[181, 181, 90, 2, 1, "7919/2172", "7919/2172", "0"],
[182, 91, -1, 1, 0, "8009/2184", "8009/2184", "0"],
[183, 183, 91, 2, 1, "4049/1098", "4049/1098", "0"],
[184, 92, -1, 1, 0, "8189/2208", "8189/2208", "0"],
[185, 185, 92, 2, 1, "8279/2220", "8279/2220", "0"],
[186, 93, -1, 1, 0, "8371/2232", "8371/2232", "0"],
[187, 187, 93, 2, 1, "4231/1122", "4231/1122", "0"],
[188, 94, -1, 1, 0, "8555/2256", "8555/2256", "0"],
[189, 189, 94, 2, 1, "8647/2268", "8647/2268", "0"],
[190, 95, -1, 1, 0, "8741/2280", "8741/2280", "0"],
[191, 191, 95, 2, 1, "4417/1146", "4417/1146", "0"],
[192, 96, -1, 1, 0, "8929/2304", "8929/2304", "0"],
[193, 193, 96, 2, 1, "9023/2316", "9023/2316", "0"],
[194, 97, -1, 1, 0, "9119/2328", "9119/2328", "0"],
[195, 195, 97, 2, 1, "4607/1170", "4607/1170", "0"],
[196, 98, -1, 1, 0, "9311/2352", "9311/2352", "0"],
[197, 197, 98, 2, 1, "9407/2364", "9407/2364", "0"],
```

```
[198, 99, -1, 1, 0, "9505/2376", "9505/2376", "0"],
[199, 199, 99, 2, 1, "4801/1194", "4801/1194", "0"],
[200, 100, -1, 1, 0, "9701/2400", "9701/2400", "0"],
[201, 201, 100, 2, 1, "9799/2412", "9799/2412", "0"],
[202, 101, -1, 1, 0, "9899/2424", "9899/2424", "0"],
[203, 203, 101, 2, 1, "4999/1218", "4999/1218", "0"],
[204, 102, -1, 1, 0, "10099/2448", "10099/2448", "0"],
[205, 205, 102, 2, 1, "10199/2460", "10199/2460", "0"],
[206, 103, -1, 1, 0, "10301/2472", "10301/2472", "0"],
[207, 207, 103, 2, 1, "5201/1242", "5201/1242", "0"],
[208, 104, -1, 1, 0, "10505/2496", "10505/2496", "0"],
[209, 209, 104, 2, 1, "10607/2508", "10607/2508", "0"],
[210, 105, -1, 1, 0, "10711/2520", "10711/2520", "0"],
[211, 211, 105, 2, 1, "5407/1266", "5407/1266", "0"],
[212, 106, -1, 1, 0, "10919/2544", "10919/2544", "0"],
[213, 213, 106, 2, 1, "11023/2556", "11023/2556", "0"],
[214, 107, -1, 1, 0, "11129/2568", "11129/2568", "0"],
[215, 215, 107, 2, 1, "5617/1290", "5617/1290", "0"],
[216, 108, -1, 1, 0, "11341/2592", "11341/2592", "0"],
[217, 217, 108, 2, 1, "11447/2604", "11447/2604", "0"],
[218, 109, -1, 1, 0, "11555/2616", "11555/2616", "0"],
[219, 219, 109, 2, 1, "5831/1314", "5831/1314", "0"],
[220, 110, -1, 1, 0, "11771/2640", "11771/2640", "0"],
[221, 221, 110, 2, 1, "11879/2652", "11879/2652", "0"],
[222, 111, -1, 1, 0, "11989/2664", "11989/2664", "0"],
[223, 223, 111, 2, 1, "6049/1338", "6049/1338", "0"],
[224, 112, -1, 1, 0, "12209/2688", "12209/2688", "0"],
[225, 225, 112, 2, 1, "12319/2700", "12319/2700", "0"],
[226, 113, -1, 1, 0, "12431/2712", "12431/2712", "0"],
[227, 227, 113, 2, 1, "6271/1362", "6271/1362", "0"],
[228, 114, -1, 1, 0, "12655/2736", "12655/2736", "0"],
[229, 229, 114, 2, 1, "12767/2748", "12767/2748", "0"],
[230, 115, -1, 1, 0, "12881/2760", "12881/2760", "0"],
[231, 231, 115, 2, 1, "6497/1386", "6497/1386", "0"],
[232, 116, -1, 1, 0, "13109/2784", "13109/2784", "0"],
[233, 233, 116, 2, 1, "13223/2796", "13223/2796", "0"],
[234, 117, -1, 1, 0, "13339/2808", "13339/2808", "0"],
[235, 235, 117, 2, 1, "6727/1410", "6727/1410", "0"],
[236, 118, -1, 1, 0, "13571/2832", "13571/2832", "0"],
[237, 237, 118, 2, 1, "13687/2844", "13687/2844", "0"],
[238, 119, -1, 1, 0, "13805/2856", "13805/2856", "0"],
[239, 239, 119, 2, 1, "6961/1434", "6961/1434", "0"],
[240, 120, -1, 1, 0, "14041/2880", "14041/2880", "0"],
[241, 241, 120, 2, 1, "14159/2892", "14159/2892", "0"],
[242, 121, -1, 1, 0, "14279/2904", "14279/2904", "0"],
[243, 243, 121, 2, 1, "7199/1458", "7199/1458", "0"],
[244, 122, -1, 1, 0, "14519/2928", "14519/2928", "0"],
[245, 245, 122, 2, 1, "14639/2940", "14639/2940", "0"],
[246, 123, -1, 1, 0, "14761/2952", "14761/2952", "0"],
[247, 247, 123, 2, 1, "7441/1482", "7441/1482", "0"],
[248, 124, -1, 1, 0, "15005/2976", "15005/2976", "0"],
[249, 249, 124, 2, 1, "15127/2988", "15127/2988", "0"],
[250, 125, -1, 1, 0, "15251/3000", "15251/3000", "0"],
[251, 251, 125, 2, 1, "7687/1506", "7687/1506", "0"],
[252, 126, -1, 1, 0, "15499/3024", "15499/3024", "0"],
[253, 253, 126, 2, 1, "15623/3036", "15623/3036", "0"],
[254, 127, -1, 1, 0, "15749/3048", "15749/3048", "0"],
[255, 255, 127, 2, 1, "7937/1530", "7937/1530", "0"],
[256, 128, -1, 1, 0, "16001/3072", "16001/3072", "0"],
```

```
[257, 257, 128, 2, 1, "16127/3084", "16127/3084", "0"],
[258, 129, -1, 1, 0, "16255/3096", "16255/3096", "0"],
[259, 259, 129, 2, 1, "8191/1554", "8191/1554", "0"],
[260, 130, -1, 1, 0, "16511/3120", "16511/3120", "0"],
[261, 261, 130, 2, 1, "16639/3132", "16639/3132", "0"],
[262, 131, -1, 1, 0, "16769/3144", "16769/3144", "0"],
[263, 263, 131, 2, 1, "8449/1578", "8449/1578", "0"],
[264, 132, -1, 1, 0, "17029/3168", "17029/3168", "0"],
[265, 265, 132, 2, 1, "17159/3180", "17159/3180", "0"],
[266, 133, -1, 1, 0, "17291/3192", "17291/3192", "0"],
[267, 267, 133, 2, 1, "8711/1602", "8711/1602", "0"],
[268, 134, -1, 1, 0, "17555/3216", "17555/3216", "0"],
[269, 269, 134, 2, 1, "17687/3228", "17687/3228", "0"],
[270, 135, -1, 1, 0, "17821/3240", "17821/3240", "0"],
[271, 271, 135, 2, 1, "8977/1626", "8977/1626", "0"],
[272, 136, -1, 1, 0, "18089/3264", "18089/3264", "0"],
[273, 273, 136, 2, 1, "18223/3276", "18223/3276", "0"],
[274, 137, -1, 1, 0, "18359/3288", "18359/3288", "0"],
[275, 275, 137, 2, 1, "9247/1650", "9247/1650", "0"],
[276, 138, -1, 1, 0, "18631/3312", "18631/3312", "0"],
[277, 277, 138, 2, 1, "18767/3324", "18767/3324", "0"],
[278, 139, -1, 1, 0, "18905/3336", "18905/3336", "0"],
[279, 279, 139, 2, 1, "9521/1674", "9521/1674", "0"],
[280, 140, -1, 1, 0, "19181/3360", "19181/3360", "0"],
[281, 281, 140, 2, 1, "19319/3372", "19319/3372", "0"],
[282, 141, -1, 1, 0, "19459/3384", "19459/3384", "0"],
[283, 283, 141, 2, 1, "9799/1698", "9799/1698", "0"],
[284, 142, -1, 1, 0, "19739/3408", "19739/3408", "0"],
[285, 285, 142, 2, 1, "19879/3420", "19879/3420", "0"],
[286, 143, -1, 1, 0, "20021/3432", "20021/3432", "0"],
[287, 287, 143, 2, 1, "10081/1722", "10081/1722", "0"],
[288, 144, -1, 1, 0, "20305/3456", "20305/3456", "0"],
[289, 289, 144, 2, 1, "20447/3468", "20447/3468", "0"],
[290, 145, -1, 1, 0, "20591/3480", "20591/3480", "0"],
[291, 291, 145, 2, 1, "10367/1746", "10367/1746", "0"],
[292, 146, -1, 1, 0, "20879/3504", "20879/3504", "0"],
[293, 293, 146, 2, 1, "21023/3516", "21023/3516", "0"],
[294, 147, -1, 1, 0, "21169/3528", "21169/3528", "0"],
[295, 295, 147, 2, 1, "10657/1770", "10657/1770", "0"],
[296, 148, -1, 1, 0, "21461/3552", "21461/3552", "0"],
[297, 297, 148, 2, 1, "21607/3564", "21607/3564", "0"],
[298, 149, -1, 1, 0, "21755/3576", "21755/3576", "0"],
[299, 299, 149, 2, 1, "10951/1794", "10951/1794", "0"],
[300, 150, -1, 1, 0, "22051/3600", "22051/3600", "0"],
[301, 301, 150, 2, 1, "22199/3612", "22199/3612", "0"],
[302, 151, -1, 1, 0, "22349/3624", "22349/3624", "0"],
[303, 303, 151, 2, 1, "11249/1818", "11249/1818", "0"],
[304, 152, -1, 1, 0, "22649/3648", "22649/3648", "0"],
[305, 305, 152, 2, 1, "22799/3660", "22799/3660", "0"],
[306, 153, -1, 1, 0, "22951/3672", "22951/3672", "0"],
[307, 307, 153, 2, 1, "11551/1842", "11551/1842", "0"],
[308, 154, -1, 1, 0, "23255/3696", "23255/3696", "0"],
[309, 309, 154, 2, 1, "23407/3708", "23407/3708", "0"],
[310, 155, -1, 1, 0, "23561/3720", "23561/3720", "0"],
[311, 311, 155, 2, 1, "11857/1866", "11857/1866", "0"],
[312, 156, -1, 1, 0, "23869/3744", "23869/3744", "0"],
[313, 313, 156, 2, 1, "24023/3756", "24023/3756", "0"],
[314, 157, -1, 1, 0, "24179/3768", "24179/3768", "0"],
[315, 315, 157, 2, 1, "12167/1890", "12167/1890", "0"],
```

```
[316, 158, -1, 1, 0, "24491/3792", "24491/3792", "0"],
[317, 317, 158, 2, 1, "24647/3804", "24647/3804", "0"],
[318, 159, -1, 1, 0, "24805/3816", "24805/3816", "0"],
[319, 319, 159, 2, 1, "12481/1914", "12481/1914", "0"],
[320, 160, -1, 1, 0, "25121/3840", "25121/3840", "0"],
[321, 321, 160, 2, 1, "25279/3852", "25279/3852", "0"],
[322, 161, -1, 1, 0, "25439/3864", "25439/3864", "0"],
[323, 323, 161, 2, 1, "12799/1938", "12799/1938", "0"],
[324, 162, -1, 1, 0, "25759/3888", "25759/3888", "0"],
[325, 325, 162, 2, 1, "25919/3900", "25919/3900", "0"],
[326, 163, -1, 1, 0, "26081/3912", "26081/3912", "0"],
[327, 327, 163, 2, 1, "13121/1962", "13121/1962", "0"],
[328, 164, -1, 1, 0, "26405/3936", "26405/3936", "0"],
[329, 329, 164, 2, 1, "26567/3948", "26567/3948", "0"],
[330, 165, -1, 1, 0, "26731/3960", "26731/3960", "0"],
[331, 331, 165, 2, 1, "13447/1986", "13447/1986", "0"],
[332, 166, -1, 1, 0, "27059/3984", "27059/3984", "0"],
[333, 333, 166, 2, 1, "27223/3996", "27223/3996", "0"],
[334, 167, -1, 1, 0, "27389/4008", "27389/4008", "0"],
[335, 335, 167, 2, 1, "13777/2010", "13777/2010", "0"],
[336, 168, -1, 1, 0, "27721/4032", "27721/4032", "0"],
[337, 337, 168, 2, 1, "27887/4044", "27887/4044", "0"],
[338, 169, -1, 1, 0, "28055/4056", "28055/4056", "0"],
[339, 339, 169, 2, 1, "14111/2034", "14111/2034", "0"],
[340, 170, -1, 1, 0, "28391/4080", "28391/4080", "0"],
[341, 341, 170, 2, 1, "28559/4092", "28559/4092", "0"],
[342, 171, -1, 1, 0, "28729/4104", "28729/4104", "0"],
[343, 343, 171, 2, 1, "14449/2058", "14449/2058", "0"],
[344, 172, -1, 1, 0, "29069/4128", "29069/4128", "0"],
[345, 345, 172, 2, 1, "29239/4140", "29239/4140", "0"],
[346, 173, -1, 1, 0, "29411/4152", "29411/4152", "0"],
[347, 347, 173, 2, 1, "14791/2082", "14791/2082", "0"],
[348, 174, -1, 1, 0, "29755/4176", "29755/4176", "0"],
[349, 349, 174, 2, 1, "29927/4188", "29927/4188", "0"],
[350, 175, -1, 1, 0, "30101/4200", "30101/4200", "0"],
[351, 351, 175, 2, 1, "15137/2106", "15137/2106", "0"],
[352, 176, -1, 1, 0, "30449/4224", "30449/4224", "0"],
[353, 353, 176, 2, 1, "30623/4236", "30623/4236", "0"],
[354, 177, -1, 1, 0, "30799/4248", "30799/4248", "0"],
[355, 355, 177, 2, 1, "15487/2130", "15487/2130", "0"],
[356, 178, -1, 1, 0, "31151/4272", "31151/4272", "0"],
[357, 357, 178, 2, 1, "31327/4284", "31327/4284", "0"],
[358, 179, -1, 1, 0, "31505/4296", "31505/4296", "0"],
[359, 359, 179, 2, 1, "15841/2154", "15841/2154", "0"],
[360, 180, -1, 1, 0, "31861/4320", "31861/4320", "0"],
[361, 361, 180, 2, 1, "32039/4332", "32039/4332", "0"],
[362, 181, -1, 1, 0, "32219/4344", "32219/4344", "0"],
[363, 363, 181, 2, 1, "16199/2178", "16199/2178", "0"],
[364, 182, -1, 1, 0, "32579/4368", "32579/4368", "0"],
[365, 365, 182, 2, 1, "32759/4380", "32759/4380", "0"],
[366, 183, -1, 1, 0, "32941/4392", "32941/4392", "0"],
[367, 367, 183, 2, 1, "16561/2202", "16561/2202", "0"],
[368, 184, -1, 1, 0, "33305/4416", "33305/4416", "0"],
[369, 369, 184, 2, 1, "33487/4428", "33487/4428", "0"],
[370, 185, -1, 1, 0, "33671/4440", "33671/4440", "0"],
[371, 371, 185, 2, 1, "16927/2226", "16927/2226", "0"],
[372, 186, -1, 1, 0, "34039/4464", "34039/4464", "0"],
[373, 373, 186, 2, 1, "34223/4476", "34223/4476", "0"],
[374, 187, -1, 1, 0, "34409/4488", "34409/4488", "0"],
```

```
      [375, 375, 187, 2, 1, "17297/2250", "17297/2250", "0"],
      [376, 188, -1, 1, 0, "34781/4512", "34781/4512", "0"],
      [377, 377, 188, 2, 1, "34967/4524", "34967/4524", "0"],
      [378, 189, -1, 1, 0, "35155/4536", "35155/4536", "0"],
      [379, 379, 189, 2, 1, "17671/2274", "17671/2274", "0"],
      [380, 190, -1, 1, 0, "35531/4560", "35531/4560", "0"],
      [381, 381, 190, 2, 1, "35719/4572", "35719/4572", "0"],
      [382, 191, -1, 1, 0, "35909/4584", "35909/4584", "0"],
      [383, 383, 191, 2, 1, "18049/2298", "18049/2298", "0"],
      [384, 192, -1, 1, 0, "36289/4608", "36289/4608", "0"],
      [385, 385, 192, 2, 1, "36479/4620", "36479/4620", "0"],
      [386, 193, -1, 1, 0, "36671/4632", "36671/4632", "0"],
      [387, 387, 193, 2, 1, "18431/2322", "18431/2322", "0"],
      [388, 194, -1, 1, 0, "37055/4656", "37055/4656", "0"],
      [389, 389, 194, 2, 1, "37247/4668", "37247/4668", "0"],
      [390, 195, -1, 1, 0, "37441/4680", "37441/4680", "0"],
      [391, 391, 195, 2, 1, "18817/2346", "18817/2346", "0"],
      [392, 196, -1, 1, 0, "37829/4704", "37829/4704", "0"],
      [393, 393, 196, 2, 1, "38023/4716", "38023/4716", "0"],
      [394, 197, -1, 1, 0, "38219/4728", "38219/4728", "0"],
      [395, 395, 197, 2, 1, "19207/2370", "19207/2370", "0"],
      [396, 198, -1, 1, 0, "38611/4752", "38611/4752", "0"],
      [397, 397, 198, 2, 1, "38807/4764", "38807/4764", "0"],
      [398, 199, -1, 1, 0, "39005/4776", "39005/4776", "0"],
      [399, 399, 199, 2, 1, "19601/2394", "19601/2394", "0"],
      [400, 200, -1, 1, 0, "39401/4800", "39401/4800", "0"]
    ]
  },
  "eta_norm_cases": [
    {
      "depth": 2,
      "odd_part": 1,
      "valid_source_units": 64,
      "trials": 7505,
      "source_coordinates": [[0, 1], [1, 1], [2, 1]],
      "target_coordinates": [[0, 1], [1, 1]],
      "columns": [
        "trial", "source_exponents", "target_exponents", "target_weight_sum",
        "target_infinity_sum", "target_reciprocal_sum"
      ],
      "accepted_cases": [
        [49, [16, 0, -16], [48, -48], 0, -48, 48],
        [131, [16, -24, 8], [0, 0], 0, 0, 0],
        [273, [16, -24, 8], [0, 0], 0, 0, 0],
        [341, [24, -24, 0], [24, -24], 0, -24, 24],
        [378, [-8, -24, 32], [-72, 72], 0, 72, -72],
        [401, [24, 24, -48], [120, -120], 0, -120, 120],
        [444, [-8, 24, -16], [24, -24], 0, -24, 24],
        [451, [8, 24, -32], [72, -72], 0, -72, 72],
        [769, [24, 24, -48], [120, -120], 0, -120, 120],
        [803, [8, 24, -32], [72, -72], 0, -72, 72],
        [858, [-8, 0, 8], [-24, 24], 0, 24, -24],
        [1068, [-16, 0, 16], [-48, 48], 0, 48, -48],
        [1293, [-8, 24, -16], [24, -24], 0, -24, 24],
        [1378, [16, 0, -16], [48, -48], 0, -48, 48],
        [1488, [0, -24, 24], [-48, 48], 0, 48, -48],
        [1672, [16, -24, 8], [0, 0], 0, 0, 0],
        [1751, [16, 0, -16], [48, -48], 0, -48, 48],
        [1793, [16, 24, -40], [96, -96], 0, -96, 96],
```

```
      [1835, [8, 24, -32], [72, -72], 0, -72, 72],
      [1880, [0, 24, -24], [48, -48], 0, -48, 48],
      [2566, [16, 0, -16], [48, -48], 0, -48, 48],
      [2616, [-24, 24, 0], [-24, 24], 0, 24, -24],
      [2645, [-16, 24, -8], [0, 0], 0, 0, 0],
      [2711, [-8, 0, 8], [-24, 24], 0, 24, -24],
      [2796, [-16, 24, -8], [0, 0], 0, 0, 0],
      [2873, [8, -24, 16], [-24, 24], 0, 24, -24],
      [2937, [16, -24, 8], [0, 0], 0, 0, 0],
      [2984, [-8, 24, -16], [24, -24], 0, -24, 24],
      [3024, [-16, 24, -8], [0, 0], 0, 0, 0],
      [3135, [8, 24, -32], [72, -72], 0, -72, 72],
      [3176, [8, -24, 16], [-24, 24], 0, 24, -24],
      [3352, [16, -24, 8], [0, 0], 0, 0, 0],
      [3392, [-16, 24, -8], [0, 0], 0, 0, 0],
      [3709, [-8, 0, 8], [-24, 24], 0, 24, -24],
      [3853, [-16, -24, 40], [-96, 96], 0, 96, -96],
      [4091, [8, 0, -8], [24, -24], 0, -24, 24],
      [4170, [-24, -24, 48], [-120, 120], 0, 120, -120],
      [4175, [24, 24, -48], [120, -120], 0, -120, 120],
      [4374, [-16, -24, 40], [-96, 96], 0, 96, -96],
      [4578, [-24, 0, 24], [-72, 72], 0, 72, -72],
      [4603, [-16, 0, 16], [-48, 48], 0, 48, -48],
      [4632, [24, 24, -48], [120, -120], 0, -120, 120],
      [4643, [8, -24, 16], [-24, 24], 0, 24, -24],
      [4830, [-16, 24, -8], [0, 0], 0, 0, 0],
      [4965, [24, -24, 0], [24, -24], 0, -24, 24],
      [5253, [-24, -24, 48], [-120, 120], 0, 120, -120],
      [5435, [16, -24, 8], [0, 0], 0, 0, 0],
      [5469, [16, -24, 8], [0, 0], 0, 0, 0],
      [5561, [-24, -24, 48], [-120, 120], 0, 120, -120],
      [5628, [16, 24, -40], [96, -96], 0, -96, 96],
      [5705, [-24, 24, 0], [-24, 24], 0, 24, -24],
      [5750, [-8, -24, 32], [-72, 72], 0, 72, -72],
      [5980, [-8, 0, 8], [-24, 24], 0, 24, -24],
      [6203, [-24, 0, 24], [-72, 72], 0, 72, -72],
      [6429, [0, 24, -24], [48, -48], 0, -48, 48],
      [6728, [24, 0, -24], [72, -72], 0, -72, 72],
      [6749, [16, 0, -16], [48, -48], 0, -48, 48],
      [6806, [24, 24, -48], [120, -120], 0, -120, 120],
      [6896, [0, 0, 0], [0, 0], 0, 0, 0],
      [7084, [24, -24, 0], [24, -24], 0, -24, 24],
      [7263, [16, 0, -16], [48, -48], 0, -48, 48],
      [7299, [8, -24, 16], [-24, 24], 0, 24, -24],
      [7443, [24, -24, 0], [24, -24], 0, -24, 24],
      [7505, [0, 24, -24], [48, -48], 0, -48, 48]
    ]
  },
  {
    "depth": 2,
    "odd_part": 3,
    "valid_source_units": 18,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [0, 3], [1, 3], [2, 3]],
    "target_coordinates": [[0, 1], [1, 1], [0, 3], [1, 3]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
```

```
    "accepted_cases": [
      [3535, [21, -5, 4, 9, -1, -28], [53, -13, 25, -65], 0, -288, 264],
      [4314, [-10, 19, 5, -2, 11, -23], [8, 20, 16, -44], 0, -168, 96],
      [4853, [-7, 10, -7, -19, 2, 21], [-1, -7, -53, 61], 0, 192, -72],
      [4934, [21, 2, -7, 1, -14, -3], [67, -35, -25, -7], 0, -120, 240],
      [5006, [5, -12, -23, 9, 0, 21], [-9, -51, 27, 33], 0, 168, -120],
      [5025, [-1, -4, 9, 11, 4, -19], [-11, 19, 41, -49], 0, -144, 24],
      [5901, [-10, -19, 3, 6, 13, 7], [-68, 16, 44, 8], 0, 144, -264],
      [7516, [9, 24, -21, 13, 24, -49], [75, -51, 87, -111], 0, -432, 360],
      [7588, [-14, 7, 3, -6, 11, -1], [-28, 20, 4, 4], 0, 48, -96],
      [7659, [19, 18, -1, 15, 18, -69], [93, -21, 81, -153], 0, -624, 504],
      [9103, [15, -17, -20, -21, -1, 44], [11, -55, -65, 109], 0, 360, -120],
      [10033, [13, 8, 7, -7, -8, -13], [55, 1, -37, -19], 0, -168, 240],
      [11723, [0, 4, 16, -16, 20, -24], [8, 32, -8, -32], 0, -144, 96],
      [13312, [-22, -7, -3, 18, -11, 25], [-80, 16, 32, 32], 0, 240, -336],
      [13509, [-17, 7, 6, 11, -13, 6], [-37, 29, 7, 1], 0, 48, -120],
      [14212, [-24, 22, -8, -24, 2, 32], [-28, 8, -68, 88], 0, 312, -192],
      [14440, [21, 5, 8, 9, 13, -56], [73, -5, 53, -121], 0, -504, 408],
      [14471, [23, 1, -10, 3, 17, -34], [71, -43, 43, -71], 0, -312, 312]
    ]
  },
  {
    "depth": 2,
    "odd_part": 5,
    "valid_source_units": 45,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [0, 5], [1, 5], [2, 5]],
    "target_coordinates": [[0, 1], [1, 1], [0, 5], [1, 5]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [340, [3, 1, 12, 9, -21, -4], [11, 21, -15, -17], 0, -192, 168],
      [585, [-3, -12, -21, 7, -12, 41], [-33, -39, -3, 75], 0, 624, -456],
      [729, [0, 13, 7, -24, 23, -19], [26, 14, -26, -14], 0, -216, 264],
      [1092, [18, -21, 19, 22, 17, -55], [12, 20, 100, -132], 0, -768, 288],
      [3586, [6, 0, -18, 18, 0, -6], [18, -42, 54, -30], 0, -96, 48],
      [3618, [3, -3, -16, 17, 7, -8], [3, -35, 65, -33], 0, -72, -48],
      [4099, [9, 0, 23, -13, -8, -11], [27, 37, -55, -9], 0, -264, 336],
      [4254, [9, 8, -19, 11, 16, -25], [43, -47, 65, -61], 0, -336, 264],
      [5053, [-22, -1, 17, 14, -19, 11], [-68, 56, 4, 8], 0, 144, -384],
      [5136, [-2, 18, 24, 10, 14, -64], [30, 50, 58, -138], 0, -960, 528],
      [5267, [24, 12, 24, 24, -12, -72], [96, 24, 48, -168], 0, -1296, 1008],
      [5328, [22, 17, -23, -6, -5, -5], [100, -68, -28, -4], 0, -216, 600],
      [6017, [-9, -3, -4, 21, 7, -12], [-33, 1, 77, -45], 0, -96, -216],
      [6239, [4, -10, -4, -4, 18, -4], [-8, -12, 24, -4], 0, 48, -96],
      [6720, [24, -19, -17, 24, 23, -35], [34, -58, 118, -94], 0, -432, 192],
      [6729, [6, -23, -1, -22, -5, 45], [-28, -8, -76, 112], 0, 696, -360],
      [6793, [1, -5, 14, 19, -15, -14], [-7, 27, 27, -47], 0, -288, 72],
      [6837, [-24, -19, 21, 8, -9, 23], [-110, 66, 6, 38], 0, 432, -720],
      [7132, [16, 19, -1, 8, 9, -51], [86, -18, 42, -110], 0, -840, 744],
      [7236, [8, -7, -23, 0, -21, 43], [10, -54, -42, 86], 0, 552, -168],
      [7609, [-2, 4, 8, -6, 12, -16], [2, 18, 6, -26], 0, -192, 96],
      [8191, [-14, -18, -2, 6, -14, 42], [-78, 10, -10, 78], 0, 672, -672],
      [8306, [24, 5, -21, 16, -9, -15], [82, -66, 30, -46], 0, -360, 504],
      [9519, [-18, 11, -7, -22, 1, 35], [-32, 4, -64, 92], 0, 576, -336],
      [9777, [-21, -3, -10, -15, 7, 42], [-69, 1, -31, 99], 0, 768, -648],
      [9887, [-12, 16, 24, -12, -24, 8], [-4, 60, -84, 28], 0, -24, 120],
```

```
      [9916, [-23, -23, -12, -13, -13, 84], [-115, -1, -65, 181], 0, 1368, -1104],
      [10674, [11, -2, -21, 17, 10, -15], [29, -53, 71, -47], 0, -192, 120],
      [10682, [-7, -10, 5, -21, 2, 31], [-41, 17, -59, 83], 0, 528, -360],
      [11040, [-17, -9, 8, -3, -3, 24], [-69, 33, -15, 51], 0, 432, -504],
      [11691, [11, 19, 14, 17, 17, -78], [71, 17, 85, -173], 0, -1200, 792],
      [11963, [-11, 22, -11, 7, 10, -17], [11, -11, 41, -41], 0, -216, 96],
      [12793, [-5, 6, -17, -7, -14, 37], [-3, -29, -49, 81], 0, 504, -192],
      [12908, [-13, -18, -23, -23, -6, 83], [-75, -33, -81, 189], 0, 1344, -888],
      [13456, [-23, -2, -9, -13, 2, 45], [-73, 5, -35, 103], 0, 792, -672],
      [13832, [-8, 6, -16, 8, 18, -8], [-12, -24, 60, -24], 0, 0, -144],
      [13838, [-10, -8, -20, -14, 24, 28], [-46, -30, 6, 70], 0, 624, -528],
      [14072, [24, 3, -9, 24, -15, -27], [78, -42, 42, -78], 0, -576, 576],
      [14304, [2, 23, -7, 14, -19, -13], [52, -16, 4, -40], 0, -360, 408],
      [15530, [-15, -5, 16, -13, -7, 24], [-55, 47, -53, 61], 0, 384, -360],
      [16871, [22, 7, 11, -14, 21, -47], [80, 0, 0, -80], 0, -720, 720],
      [17281, [15, 24, 7, -3, -16, -27], [93, -1, -41, -51], 0, -624, 792],
      [17519, [-17, -9, -12, 13, 5, 20], [-69, -7, 49, 27], 0, 432, -600],
      [17621, [-6, -21, -11, -18, -7, 63], [-60, -16, -68, 144], 0, 1008, -672],
      [17759, [-10, -16, 10, -22, -24, 62], [-62, 30, -114, 146], 0, 888, -552]
    ]
  },
  {
    "depth": 2,
    "odd_part": 9,
    "valid_source_units": 16,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [0, 3], [1, 3], [2, 3], [0, 9], [1, 9],
[2, 9]],
    "target_coordinates": [[0, 1], [1, 1], [0, 3], [1, 3], [0, 9], [1, 9]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        415,
        [17, -7, 15, 14, -17, -11, -19, 14, -6],
        [37, 13, 8, -36, -29, 7],
        0,
        -264,
        672
      ],
      [
        1054,
        [20, 23, 18, 19, 4, 7, -5, 1, -87],
        [106, 16, 65, -5, -13, -169],
        0,
        -2856,
        2232
      ],
      [
        1144,
        [3, -20, -5, 5, -8, -1, -10, -20, 56],
        [-31, -13, -1, -7, -70, 122],
        0,
        1464,
        -720
      ],
      [
```

```
  1341,
  [-12, 24, 6, -16, -3, 7, 12, -15, -3],
  [12, 24, -54, 30, 6, -18],
  0,
  -192,
  192
],
[
  2561,
  [-18, 14, -16, 0, 20, 22, 10, -14, -18],
  [-26, -14, 40, 44, 2, -46],
  0,
  -480,
  -264
],
[
  3344,
  [15, -21, -6, -7, -5, -6, 22, 24, -16],
  [3, -27, -31, -5, 114, -54],
  0,
  -120,
  -216
],
[
  3781,
  [15, 10, 10, 5, 19, -8, 10, -3, -58],
  [65, 5, 53, -21, 24, -126],
  0,
  -1944,
  1392
],
[
  6749,
  [-17, 18, 8, -17, -10, -23, -20, 16, 45],
  [-15, 33, -71, -29, -28, 110],
  0,
  1392,
  -432
],
[
  8719,
  [-17, -2, 24, 3, 21, -6, 24, -1, -46],
  [-55, 65, 51, -15, 70, -116],
  0,
  -1320,
  -120
],
[
  9166,
  [-19, 4, 17, 13, 6, 1, -12, -14, 4],
  [-49, 53, 51, -11, -64, 20],
  0,
  -72,
  -240
],
[
  9354,
  [-3, 4, -5, -11, 7, -20, -4, 23, 9],
  [-1, -7, -19, -29, 34, 22],
```

```
        0,
        456,
        -192
      ],
      [
        10141,
        [-5, 24, -19, 4, 21, 17, 17, -3, -56],
        [33, -33, 54, 30, 45, -129],
        0,
        -1608,
        672
      ],
      [
        11217,
        [-17, 20, 1, -3, -2, -19, -22, -14, 56],
        [-11, 19, -13, -35, -94, 134],
        0,
        1344,
        -264
      ],
      [
        11218,
        [5, -1, 0, 24, 2, -12, -21, 15, -12],
        [13, -5, 76, -48, -33, -3],
        0,
        -408,
        432
      ],
      [
        11608,
        [-14, -1, -20, -8, -22, -22, -10, 3, 94],
        [-44, -26, -68, -36, -24, 198],
        0,
        2832,
        -1392
      ],
      [
        13602,
        [-14, -4, -2, 0, 21, 15, 14, -11, -19],
        [-50, 10, 42, 30, 20, -52],
        0,
        -480,
        -480
      ]
    ]
  },
  {
    "depth": 2,
    "odd_part": 15,
    "valid_source_units": 5,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [0, 5],
```

```
      [1, 5],
      [2, 5],
      [0, 15],
      [1, 15],
      [2, 15]
    ],
    "target_coordinates": [[0, 1], [1, 1], [0, 3], [1, 3], [0, 5], [1, 5], [0, 15], [1,
15]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        4858,
        [23, 14, 12, 16, 14, 19, -11, -16, -2, -8, 0, -61],
        [97, 1, 76, 22, -65, 7, -24, -114],
        0,
        -3576,
        3264
      ],
      [
        9505,
        [-23, 3, -19, 19, -12, -10, 10, -1, 4, -20, 22, 27],
        [-63, -15, 33, -39, 28, -2, -16, 74],
        0,
        1872,
        -1776
      ],
      [
        12948,
        [-16, 1, 17, -4, -20, -4, 2, -20, -22, -2, -3, 71],
        [-46, 50, -52, -4, -34, -46, -12, 144],
        0,
        3384,
        -1392
      ],
      [
        13918,
        [22, 0, 18, -10, 4, -6, -2, -7, 23, 22, 13, -77],
        [66, 14, -22, -2, -20, 48, 92, -176],
        0,
        -3504,
        1992
      ],
      [
        15600,
        [14, -5, 10, 4, 9, -10, -23, -10, 6, -1, -20, 26],
        [32, 6, 30, -24, -89, 35, -43, 53],
        0,
        840,
        768
      ]
    ]
  },
  {
    "depth": 3,
    "odd_part": 1,
    "valid_source_units": 41,
```

```
  "trials": 18000,
  "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1]],
  "target_coordinates": [[0, 1], [1, 1], [2, 1]],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [18, [20, -2, 10, -28], [56, 0, -56], 0, -168, 168],
    [802, [-16, -8, 16, 8], [-64, 48, 16], 0, 96, -144],
    [886, [4, -6, -10, 12], [0, -24, 24], 0, 48, -24],
    [1407, [12, -6, -18, 12], [24, -48, 24], 0, 24, 24],
    [1789, [4, -2, -22, 20], [8, -48, 40], 0, 72, -24],
    [2221, [-24, -8, 0, 32], [-88, 24, 64], 0, 216, -240],
    [2308, [8, -24, -8, 24], [-24, -24, 48], 0, 120, -96],
    [2333, [4, 2, 14, -20], [16, 24, -40], 0, -96, 72],
    [3787, [-12, -10, -6, 28], [-56, 0, 56], 0, 168, -168],
    [4428, [24, -8, 0, -16], [56, -24, -32], 0, -120, 144],
    [4838, [4, -22, 14, 4], [-32, 24, 8], 0, 48, -72],
    [4911, [24, -24, 24, -24], [24, 24, -48], 0, -120, 96],
    [5114, [24, -8, 0, -16], [56, -24, -32], 0, -120, 144],
    [5394, [-20, -14, -10, 44], [-88, 0, 88], 0, 264, -264],
    [5654, [-8, 0, -16, 24], [-24, -24, 48], 0, 120, -96],
    [5852, [-8, 12, -4, 0], [0, 0, 0], 0, 0, 0],
    [6685, [-8, -8, -16, 32], [-40, -24, 64], 0, 168, -144],
    [7287, [8, 0, 16, -24], [24, 24, -48], 0, -120, 96],
    [7881, [-24, 16, 24, -16], [-40, 72, -32], 0, -24, -48],
    [8065, [4, -10, -22, 28], [-8, -48, 56], 0, 120, -72],
    [8431, [-8, -24, -16, 48], [-72, -24, 96], 0, 264, -240],
    [8837, [24, -16, 24, -32], [40, 24, -64], 0, -168, 144],
    [9152, [-4, 6, 10, -12], [0, 24, -24], 0, -48, 24],
    [9200, [-8, -4, 20, -8], [-32, 48, -16], 0, 0, -48],
    [10135, [20, 14, -14, -20], [88, -48, -40], 0, -168, 216],
    [10378, [12, -6, -18, 12], [24, -48, 24], 0, 24, 24],
    [11227, [-12, -22, -18, 52], [-80, -24, 104], 0, 288, -264],
    [11679, [-16, 20, -20, 16], [-8, -24, 32], 0, 72, -48],
    [11755, [-20, -10, -22, 52], [-80, -24, 104], 0, 288, -264],
    [12013, [20, 6, 10, -36], [72, 0, -72], 0, -216, 216],
    [12556, [-4, 14, -14, 4], [16, -24, 8], 0, 0, 24],
    [13371, [-4, -2, 10, -4], [-16, 24, -8], 0, 0, -24],
    [13917, [-4, -18, 10, 12], [-48, 24, 24], 0, 96, -120],
    [13927, [0, 20, -12, -8], [40, -24, -16], 0, -72, 96],
    [13983, [0, 8, 0, -8], [16, 0, -16], 0, -48, 48],
    [15062, [16, 8, -16, -8], [64, -48, -16], 0, -96, 144],
    [15173, [16, -24, 8, 0], [0, 0, 0], 0, 0, 0],
    [16086, [-4, -18, 10, 12], [-48, 24, 24], 0, 96, -120],
    [16362, [0, -12, -12, 24], [-24, -24, 48], 0, 120, -96],
    [16761, [4, -10, 2, 4], [-8, 0, 8], 0, 24, -24],
    [17239, [-4, -10, -14, 28], [-32, -24, 56], 0, 144, -120]
  ]
},
{
  "depth": 3,
  "odd_part": 3,
  "valid_source_units": 9,
  "trials": 18000,
  "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [0, 3], [1, 3], [2, 3], [3, 3]],
  "target_coordinates": [[0, 1], [1, 1], [2, 1], [0, 3], [1, 3], [2, 3]],
  "columns": [
```

```
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [
      362,
      [15, -21, 22, 10, -9, -15, -16, 14],
      [3, 29, 20, -57, -23, 28],
      0,
      168,
      24
    ],
    [
      2697,
      [-13, 14, 16, -5, -13, -16, -2, 19],
      [-11, 45, -10, -71, 9, 38],
      0,
      336,
      -120
    ],
    [
      3110,
      [1, 4, -19, -4, -23, -22, 23, 40],
      [11, -39, -8, -113, 69, 80],
      0,
      936,
      -360
    ],
    [
      6711,
      [-19, -14, 11, -18, -23, -6, -17, 86],
      [-85, 41, -36, -81, -11, 172],
      0,
      1608,
      -1056
    ],
    [7432, [-13, -2, 9, 4, 19, -4, -21, 8], [-43, 31, 8, 49, -61, 16], 0, 24, -216],
    [
      8818,
      [11, 19, 11, 9, -13, 21, -3, -55],
      [71, 11, 18, 3, 7, -110],
      0,
      -1104,
      888
    ],
    [
      9028,
      [22, -21, 15, 22, -18, -9, -1, -10],
      [24, 8, 44, -72, 16, -20],
      0,
      -144,
      192
    ],
    [
      17050,
      [-14, 15, -21, -20, 2, 7, -21, 52],
      [-12, -28, -40, 20, -44, 104],
      0,
      816,
```

```
        -336
      ],
      [
        17406,
        [-13, 20, -7, -22, -21, 14, 23, 6],
        [1, -1, -44, -35, 67, 12],
        0,
        264,
        -120
      ]
    ]
  },
  {
    "depth": 3,
    "odd_part": 5,
    "valid_source_units": 21,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [0, 5], [1, 5], [2, 5], [3, 5]],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [0, 5], [1, 5], [2, 5]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [49, [-4, -11, -2, 17, -16, -3, 4, 15], [-34, 0, 34, -54, 24, 30], 0, 672, -648],
      [525, [15, 6, -14, 3, 5, 10, -2, -23], [57, -43, 6, 35, -9, -46], 0, -840, 816],
      [
        2557,
        [20, 14, -5, -5, -16, -24, 19, -3],
        [88, -30, -10, -96, 54, -6],
        0,
        -72,
        1128
      ],
      [
        3279,
        [12, 22, -13, -5, -8, -16, -21, 29],
        [80, -38, -10, -56, -34, 58],
        0,
        504,
        936
      ],
      [
        3354,
        [17, 22, -2, -9, -13, 18, 18, -51],
        [95, -21, -18, -3, 49, -102],
        0,
        -1584,
        1584
      ],
      [
        3367,
        [8, -14, -16, -6, 8, -18, 0, 38],
        [-4, -40, -12, -12, -8, 76],
        0,
        1248,
        -528
      ],
      [
```

```
  3942,
  [-9, -18, -14, 9, 1, 0, -16, 47],
  [-63, -19, 18, 3, -33, 94],
  0,
  1536,
  -1320
],
[
  4012,
  [-16, -3, 8, 19, -20, 9, 22, -19],
  [-54, 32, 38, -42, 64, -38],
  0,
  -168,
  -648
],
[
  7391,
  [-24, -23, -8, 17, -12, 1, 18, 31],
  [-118, 8, 34, -34, 48, 62],
  0,
  1584,
  -2088
],
[7596, [-4, -15, 0, 11, -12, 15, 4, 1], [-42, 4, 22, -6, 20, 2], 0, 264, -672],
[
  8530,
  [23, 7, 22, 6, -11, -11, 10, -46],
  [83, 21, 12, -55, 31, -92],
  0,
  -1632,
  1680
],
[
  9743,
  [-24, -3, 22, -5, 8, 3, -18, 17],
  [-78, 68, -10, 30, -44, 34],
  0,
  408,
  -864
],
[
  12388,
  [18, -21, 15, -18, 14, -7, -11, 10],
  [12, 12, -36, 28, -36, 20],
  0,
  72,
  240
],
[
  12750,
  [-17, 7, -13, 7, -23, -13, -21, 73],
  [-37, -9, 14, -95, -19, 146],
  0,
  2256,
  -1032
],
[
  13459,
  [17, -18, -11, 6, -9, -20, 21, 14],
```

```
      [15, -39, 12, -67, 51, 28],
      0,
      720,
      -168
    ],
    [
      13768,
      [13, 8, -19, 20, 7, 12, -5, -36],
      [55, -51, 40, 45, -17, -72],
      0,
      -1272,
      864
    ],
    [
      13917,
      [17, 14, 20, 13, -5, -2, 8, -65],
      [79, 23, 26, -19, 21, -130],
      0,
      -2256,
      1776
    ],
    [
      15108,
      [4, 16, -20, 16, 16, -10, 6, -28],
      [44, -44, 32, 28, -4, -56],
      0,
      -936,
      648
    ],
    [
      15369,
      [-3, -16, -3, -8, 23, -20, -5, 32],
      [-41, -3, -16, 29, -33, 64],
      0,
      984,
      -816
    ],
    [
      15448,
      [-16, 0, -17, 3, -8, -20, 17, 41],
      [-48, -18, 6, -64, 42, 82],
      0,
      1680,
      -1200
    ],
    [
      15604,
      [0, -12, 8, 8, -16, -20, 8, 24],
      [-24, 16, 16, -88, 32, 48],
      0,
      912,
      -480
    ]
  ]
},
{
  "depth": 3,
  "odd_part": 9,
  "valid_source_units": 15,
```

```
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9]
    ],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [0, 3], [1, 3], [2, 3], [0, 9], [1, 9],
[2, 9]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        69,
        [23, -20, -23, 20, 12, 18, 14, 0, 1, 12, 7, -64],
        [29, -69, 40, 72, 16, 0, 27, 13, -128],
        0,
        -3768,
        1128
      ],
      [
        4593,
        [5, 9, 21, 8, 10, 22, -12, 18, 21, 5, -21, -86],
        [33, 37, 16, 74, -34, 36, 73, -63, -172],
        0,
        -6048,
        2784
      ],
      [
        6956,
        [3, 13, 1, -9, 1, -14, -23, 20, 6, 19, -2, -15],
        [35, -1, -18, -25, -47, 40, 56, -10, -30],
        0,
        -672,
        792
      ],
      [
        7568,
        [18, 7, 0, -22, -22, 21, 16, -1, 12, 8, 14, -51],
        [68, -18, -44, -24, 54, -2, 52, 16, -102],
        0,
        -2832,
        1896
      ],
      [
        7952,
        [1, -1, -20, -9, 6, 0, 12, 16, 13, -23, -20, 25],
        [1, -41, -18, 18, 18, 32, -7, -53, 50],
        0,
```

```
  1176,
  -528
],
[
  8038,
  [1, -12, 4, -22, -5, -13, 12, -8, 22, 3, 4, 14],
  [-21, 7, -44, -41, 29, -16, 72, -14, 28],
  0,
  1080,
  -1104
],
[
  9010,
  [11, 24, -2, 12, -20, -1, 9, 2, 9, 3, 11, -58],
  [81, -15, 24, -62, 38, 4, 33, 13, -116],
  0,
  -3408,
  2400
],
[
  9214,
  [-7, -19, -9, -3, -21, 22, 20, 1, 2, -1, -21, 36],
  [-59, -11, -6, -19, 61, 2, 4, -44, 72],
  0,
  2064,
  -2232
],
[
  10128,
  [21, 18, 22, 22, -18, 12, 9, 7, -7, -2, 7, -91],
  [99, 23, 44, -30, 36, 14, -25, 21, -182],
  0,
  -5784,
  4032
],
[
  10754,
  [7, -4, 6, -7, -15, -21, 18, -6, 10, -15, -2, 29],
  [13, 5, -14, -87, 51, -12, 0, -14, 58],
  0,
  1704,
  -312
],
[
  11405,
  [19, -21, 22, -13, 22, -7, -16, -5, 7, 24, -24, -8],
  [15, 25, -26, 52, -54, -10, 69, -55, -16],
  0,
  -1272,
  1176
],
[
  12697,
  [-4, -9, -19, 13, -21, -21, 16, -2, -9, -14, -23, 93],
  [-30, -34, 26, -105, 53, -4, -55, -37, 186],
  0,
  5496,
  -2520
],
```

```
        [
          13494,
          [-20, -22, 7, 3, -12, -14, 21, -5, 20, 10, -20, 32],
          [-104, 34, 6, -64, 54, -10, 80, -60, 64],
          0,
          1944,
          -3288
        ],
        [
          15683,
          [-6, -23, -16, 16, 3, -12, -8, 1, -7, 5, 22, 25],
          [-64, -26, 32, -15, -19, 2, -11, 51, 50],
          0,
          2496,
          -2664
        ],
        [
          17416,
          [23, -9, -9, -16, 18, 4, 0, 24, -21, -15, 17, -16],
          [51, -41, -32, 62, -18, 48, -93, 55, -32],
          0,
          -504,
          1296
        ]
      ]
    },
    {
      "depth": 3,
      "odd_part": 15,
      "valid_source_units": 2,
      "trials": 18000,
      "source_coordinates": [
        [0, 1],
        [1, 1],
        [2, 1],
        [3, 1],
        [0, 3],
        [1, 3],
        [2, 3],
        [3, 3],
        [0, 5],
        [1, 5],
        [2, 5],
        [3, 5],
        [0, 15],
        [1, 15],
        [2, 15],
        [3, 15]
      ],
      "target_coordinates": [
        [0, 1],
        [1, 1],
        [2, 1],
        [0, 3],
        [1, 3],
        [2, 3],
        [0, 5],
        [1, 5],
        [2, 5],
```

```
      [0, 15],
      [1, 15],
      [2, 15]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        824,
        [-19, 17, -16, -16, 18, 23, 21, 2, -10, 23, 10, -23, 13, 19, 3, -65],
        [-23, -13, -32, 100, 24, 4, 16, 30, -46, 77, -7, -130],
        0,
        -7080,
        408
      ],
      [
        6761,
        [-8, 12, 7, 24, 5, 18, 24, -6, 6, 12, -13, -6, 21, -18, 10, -88],
        [0, 22, 48, 51, 43, -12, 42, -32, -12, 27, -1, -176],
        0,
        -10032,
        2976
      ]
    ]
  },
  {
    "depth": 4,
    "odd_part": 1,
    "valid_source_units": 31,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1]],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [979, [-16, -4, 10, 22, -12], [-56, 36, 44, -24], 0, 0, -240],
      [2020, [12, -10, -20, 22, -4], [16, -52, 44, -8], 0, 24, 0],
      [3192, [-2, -23, 8, 23, -6], [-52, 18, 46, -12], 0, 72, -264],
      [4807, [20, 2, -18, -20, 16], [64, -56, -40, 32], 0, 48, 240],
      [5274, [24, 20, 20, -8, -56], [112, 16, -16, -112], 0, -816, 816],
      [5438, [4, 22, -10, -16, 0], [56, -24, -32, 0], 0, -120, 288],
      [5688, [-12, -22, -10, 4, 40], [-80, -8, 8, 80], 0, 576, -576],
      [6945, [8, 0, -2, 6, -12], [24, -12, 12, -24], 0, -144, 144],
      [7580, [20, -18, 14, 0, -16], [24, 8, 0, -32], 0, -216, 192],
      [7800, [-10, -23, 16, -1, 18], [-76, 42, -2, 36], 0, 288, -408],
      [8292, [16, -8, 8, -16, 0], [32, 0, -32, 0], 0, -96, 192],
      [8365, [4, 10, 6, -4, -16], [32, 8, -8, -32], 0, -240, 240],
      [9523, [20, 10, -14, -16, 0], [80, -48, -32, 0], 0, -144, 384],
      [9958, [24, -8, -20, -4, 8], [56, -64, -8, 16], 0, 24, 192],
      [11801, [-20, 22, 12, 14, -28], [-16, 44, 28, -56], 0, -264, 48],
      [11845, [-6, -9, 16, 9, -10], [-36, 38, 18, -20], 0, -48, -120],
      [12276, [-12, -18, -22, 12, 40], [-72, -32, 24, 80], 0, 600, -576],
      [12573, [-12, 22, 6, -16, 0], [8, 24, -32, 0], 0, -72, 96],
      [13056, [-2, 17, 8, 7, -30], [28, 18, 14, -60], 0, -360, 264],
      [13129, [22, 5, -18, 1, -10], [76, -58, 2, -20], 0, -192, 360],
```

```
      [14285, [2, 11, -18, 19, -14], [28, -38, 38, -28], 0, -120, 120],
      [14740, [-2, -3, -6, -3, 14], [-12, -10, -6, 28], 0, 168, -120],
      [14957, [-8, 4, -8, -4, 16], [-16, -8, -8, 32], 0, 192, -144],
      [15745, [20, -14, 22, -4, -24], [32, 24, -8, -48], 0, -336, 288],
      [16069, [20, 22, 10, -4, -48], [104, 0, -8, -96], 0, -696, 720],
      [16164, [10, 19, 0, 5, -34], [68, -10, 10, -68], 0, -456, 456],
      [16441, [-12, 18, 10, 24, -40], [0, 32, 48, -80], 0, -384, 144],
      [16495, [10, 7, 2, -13, -6], [44, -6, -26, -12], 0, -168, 264],
      [16841, [0, 20, 12, 16, -48], [40, 24, 32, -96], 0, -552, 384],
      [17253, [-8, 16, 14, -10, -12], [8, 36, -20, -24], 0, -192, 144],
      [17850, [4, -10, -24, -14, 44], [-8, -52, -28, 88], 0, 480, -240]
    ]
  },
  {
    "depth": 4,
    "odd_part": 3,
    "valid_source_units": 3,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [0, 3], [1, 3], [2, 3],
[3, 3], [4, 3]],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [0, 3], [1, 3], [2, 3], [3, 3]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        114,
        [17, 11, 7, -16, 13, 9, -15, -23, -4, 1],
        [73, -3, -32, 26, -3, -55, -8, 2],
        0,
        -240,
        1344
      ],
      [
        2102,
        [-15, -16, 10, -10, -15, -1, -13, -22, 15, 67],
        [-77, 35, -20, -30, -29, -43, 30, 134],
        0,
        2904,
        -1848
      ],
      [
        10312,
        [4, 14, 10, 0, 12, 8, 16, 22, -6, -80],
        [40, 16, 0, 24, 56, 36, -12, -160],
        0,
        -3336,
        1632
      ]
    ]
  },
  {
    "depth": 4,
    "odd_part": 5,
    "valid_source_units": 19,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [0, 5], [1, 5], [2, 5],
[3, 5], [4, 5]],
```

```
"target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [0, 5], [1, 5], [2, 5], [3, 5]],
"columns": [
  "trial", "source_exponents", "target_exponents", "target_weight_sum",
  "target_infinity_sum", "target_reciprocal_sum"
],
"accepted_cases": [
  [
    23,
    [-6, -20, 13, -15, 2, -4, -19, -1, 10, 40],
    [-58, 32, -30, 4, -50, 2, 20, 80],
    0,
    3288,
    -2232
  ],
  [
    747,
    [4, -18, -6, -15, -15, -6, 19, 18, -2, 21],
    [-24, -16, -30, -30, 20, 42, -4, 42],
    0,
    1704,
    -1368
  ],
  [
    1545,
    [15, -17, 20, -19, 17, 23, -20, 4, -8, -15],
    [11, 25, -38, 34, 29, -15, -16, -30],
    0,
    -1344,
    840
  ],
  [
    3503,
    [-3, 20, -4, 11, -10, 13, -23, 10, 10, -24],
    [31, -5, 22, -20, -7, 7, 20, -48],
    0,
    -1536,
    1224
  ],
  [
    4812,
    [-3, 22, -15, 21, 21, 11, 16, -5, 17, -85],
    [35, -27, 42, 42, 65, -21, 34, -170],
    0,
    -5520,
    1824
  ],
  [
    6496,
    [-3, -10, -10, -13, 6, -13, 0, -12, -11, 66],
    [-29, -17, -26, 12, -39, -11, -22, 132],
    0,
    4464,
    -1968
  ],
  [
    6791,
    [-13, -21, 8, 2, -10, 21, -5, -14, 14, 18],
    [-81, 29, 4, -20, 53, -49, 28, 36],
    0,
```

```
  1608,
  -2400
],
[
  6945,
  [-5, -21, -21, -11, 24, -3, 7, 1, 1, 28],
  [-57, -37, -22, 48, 5, 5, 2, 56],
  0,
  2520,
  -2880
],
[
  9526,
  [24, -1, 17, 22, -16, 6, -10, -7, -9, -26],
  [70, 10, 44, -32, -2, -20, -18, -52],
  0,
  -2640,
  3096
],
[
  9859,
  [-16, -24, -15, -11, 10, -18, -7, 13, -8, 76],
  [-96, -14, -22, 20, -68, 44, -16, 152],
  0,
  5808,
  -4488
],
[
  10169,
  [-23, -17, -14, -22, -18, -15, 22, 22, -3, 68],
  [-103, -5, -44, -36, -1, 59, -6, 136],
  0,
  5328,
  -4488
],
[
  10773,
  [-9, 20, -5, 24, 6, 7, 15, -3, 17, -72],
  [13, -1, 48, 12, 51, -13, 34, -144],
  0,
  -4656,
  1320
],
[
  10797,
  [6, -12, -23, -6, 13, 10, -8, 17, 0, 3],
  [-6, -52, -12, 26, 14, 24, 0, 6],
  0,
  600,
  -1056
],
[
  13628,
  [-17, 1, -13, -15, 14, -21, 20, -7, -22, 60],
  [-49, -9, -30, 28, -23, 7, -44, 120],
  0,
  3912,
  -2424
],
```

```
      [
        13744,
        [17, -8, -23, 14, 20, 11, -4, -21, 14, -20],
        [35, -63, 28, 40, 25, -53, 28, -40],
        0,
        -1104,
        624
      ],
      [
        15568,
        [-4, 2, -1, -17, 22, 18, 11, -7, 20, -44],
        [-8, 2, -34, 44, 76, -32, 40, -88],
        0,
        -2448,
        72
      ],
      [
        15848,
        [-11, 0, -23, 2, -12, 9, -17, 23, 3, 26],
        [-33, -35, 4, -24, -7, 37, 6, 52],
        0,
        2256,
        -1944
      ],
      [
        15985,
        [0, -13, -4, 22, -9, -4, -13, -10, 2, 29],
        [-26, -8, 44, -18, -38, -16, 4, 58],
        0,
        2040,
        -1152
      ],
      [
        16007,
        [-13, -10, 3, -21, 21, -17, -21, -13, -14, 85],
        [-59, 19, -42, 42, -93, -9, -28, 170],
        0,
        5832,
        -2856
      ]
    ]
  },
  {
    "depth": 4,
    "odd_part": 9,
    "valid_source_units": 9,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [0, 9],
```

```
      [1, 9],
      [2, 9],
      [3, 9],
      [4, 9]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        824,
        [-15, 24, -14, -3, -19, -23, 0, -23, 12, -8, -8, 6, 19, -3, 55],
        [3, -13, -6, -38, -69, -23, 24, -16, -12, 46, -6, 110],
        0,
        7632,
        -2352
      ],
      [
        1155,
        [5, 17, 14, 24, -8, 23, -2, 22, -12, 9, 18, 15, -18, 8, -115],
        [49, 23, 48, -16, 65, 21, -24, 18, 84, -54, 16, -230],
        0,
        -15576,
        7056
      ],
      [
        3120,
        [-17, -19, 0, -18, -5, 14, -4, -23, 19, -16, 9, 14, 21, 22, 3],
        [-89, 17, -36, -10, 34, -60, 38, -32, 55, 33, 44, 6],
        0,
        2256,
        -5640
      ],
      [
        5984,
        [-9, -7, 7, -7, -6, 24, 2, -3, 21, 14, -15, 13, -8, 14, -40],
        [-41, 23, -14, -12, 76, -30, 42, 28, -19, -1, 28, -80],
        0,
        -3864,
        -864
      ],
      [
        7648,
        [16, 9, 15, 12, -16, -4, 3, 24, 2, -3, 14, 21, -1, 1, -93],
```

```
        [66, 14, 24, -32, -6, 52, 4, -6, 84, -16, 2, -186],
        0,
        -12720,
        6312
      ],
      [
        8415,
        [-9, -13, 15, -9, 14, -4, 21, -12, -22, -13, -7, 24, 1, -23, 37],
        [-53, 39, -18, 28, 30, -20, -44, -26, 27, 9, -46, 74],
        0,
        3072,
        -2112
      ],
      [
        11234,
        [20, -21, 22, 4, 7, 0, 4, -6, -5, -5, -18, 24, 6, -10, -22],
        [18, 24, 8, 14, 8, -12, -10, -10, -6, 30, -20, -44],
        0,
        -3600,
        2376
      ],
      [
        16551,
        [-1, 3, -20, -2, 7, 14, 14, 8, -12, 0, 9, 22, -20, 19, -41],
        [3, -39, -4, 14, 70, 2, -24, 0, 71, -49, 38, -82],
        0,
        -4824,
        792
      ],
      [
        17458,
        [2, 6, -17, -17, -5, -22, -8, 15, -21, -8, 12, -8, 4, 2, 65],
        [18, -36, -34, -10, -82, 52, -42, -16, 20, -4, 4, 130],
        0,
        8520,
        -2064
      ]
    ]
  },
  {
    "depth": 4,
    "odd_part": 15,
    "valid_source_units": 6,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
```

```
      [4, 5],
      [0, 15],
      [1, 15],
      [2, 15],
      [3, 15],
      [4, 15]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [0, 15],
      [1, 15],
      [2, 15],
      [3, 15]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        3610,
        [
          10, -16, 6, -10, -11, 9, 21, 9, 10, 18, -21, -15, 7, 0, 18, -24, 18, -2, 16,
          -43
        ],
        [-2, 2, -20, -22, 69, 9, 20, 36, -93, 35, 0, 36, -36, 20, 32, -86],
        0,
        -5904,
        336
      ],
      [
        4723,
        [
          4, 9, -2, -9, -12, -16, -10, -2, -10, 8, 23, 12, 23, 8, -24, 23, -9, 5, -21,
          0
        ],
        [30, -8, -18, -24, -68, 12, -20, 16, 93, 23, 16, -48, 51, -13, -42, 0],
        0,
        -3288,
        2352
      ],
      [
        7606,
        [
          13, 6, -24, 4, -13, 23, 4, 9, -10, 6, -24, -17, 1, 9, -13, 0, 18, 10, -12,
          10
        ],
        [51, -61, 8, -26, 77, -5, -20, 12, -106, 26, 18, -26, 36, 20, -24, 20],
```

```
      0,
      1152,
      3288
    ],
    [
      11480,
      [
        2, 11, 9, -7, 6, -8, 12, -12, -20, -19, -20, -8, -15, -15, 19, 18, -15, 14,
        -24, 72
      ],
      [28, 16, -14, 12, 0, -16, -40, -38, -76, -10, -30, 38, 24, 10, -48, 144],
      0,
      14112,
      1440
    ],
    [
      13212,
      [
        -1, 9, -23, 14, -24, -2, -1, 7, 15, -16, 15, 16, -11, -5, 2, 2, 19, -13, 17,
        -20
      ],
      [15, -45, 28, -48, -8, 16, 30, -32, 77, -37, -10, 4, 44, -28, 34, -40],
      0,
      -3648,
      984
    ],
    [
      14460,
      [
        -12, -16, 10, 13, -2, 11, -13, -20, -9, 20, 15, -5, 2, 3, 16, 22, 17, 6,
        -10, -48
      ],
      [-68, 32, 26, -4, 7, -51, -18, 40, 35, -11, 6, 32, 100, -10, -20, -96],
      0,
      -9528,
      -4776
    ]
  ]
},
{
  "depth": 5,
  "odd_part": 1,
  "valid_source_units": 52,
  "trials": 18000,
  "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1]],
  "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1]],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [194, [24, 24, -18, -8, 6, -28], [120, -60, -16, 12, -56], 0, -864, 1344],
    [568, [-12, -16, 15, 20, 15, -22], [-68, 42, 40, 30, -44], 0, -288, -576],
    [858, [20, -6, 14, 8, -4, -32], [48, 8, 16, -8, -64], 0, -960, 816],
    [1147, [-6, 21, 15, -15, 3, -18], [24, 36, -30, 6, -36], 0, -552, 528],
    [1254, [-12, 12, -19, 2, -17, 34], [-12, -26, 4, -34, 68], 0, 768, -384],
    [1309, [-20, 6, 4, -10, 4, 16], [-48, 28, -20, 8, 32], 0, 504, -576],
    [2830, [10, -1, 18, 5, -4, -28], [28, 26, 10, -8, -56], 0, -840, 624],
    [2862, [-14, -11, 23, -21, -3, 26], [-64, 60, -42, -6, 52], 0, 672, -672],
```

```
      [3220, [0, -16, -12, -18, 18, 28], [-32, -24, -36, 36, 56], 0, 960, -720],
      [3747, [6, -11, -24, 15, 6, 8], [-4, -54, 30, 12, 16], 0, 360, -336],
      [4223, [-22, -3, 13, 21, 21, -30], [-72, 48, 42, 42, -60], 0, -432, -576],
      [4447, [-20, -2, -4, -14, 24, 16], [-64, 12, -28, 48, 32], 0, 744, -912],
      [4689, [-6, 23, 0, 19, 0, -36], [28, 6, 38, 0, -72], 0, -960, 576],
      [4742, [10, -5, 10, 9, -20, -4], [20, 10, 18, -40, -8], 0, -336, 384],
      [4891, [14, 3, -23, 13, -21, 14], [48, -60, 26, -42, 28], 0, 144, 336],
      [5034, [4, -2, -2, -12, 4, 8], [8, -8, -24, 8, 16], 0, 216, 0],
      [5532, [6, -7, -22, 3, -8, 28], [4, -50, 6, -16, 56], 0, 696, -288],
      [5971, [24, -18, 9, -22, -15, 22], [36, -6, -44, -30, 44], 0, 312, 336],
      [6070, [4, 8, 17, 4, 9, -42], [28, 30, 8, 18, -84], 0, -1080, 672],
      [6326, [2, -7, -21, -11, 7, 30], [-8, -44, -22, 14, 60], 0, 888, -480],
      [6824, [-6, 15, -20, -17, -16, 44], [12, -34, -34, -32, 88], 0, 960, -192],
      [7166, [16, 14, -1, -10, 3, -22], [76, -18, -20, 6, -44], 0, -696, 960],
      [7553, [4, -14, 16, 12, 10, -28], [-16, 28, 24, 20, -56], 0, -600, 48],
      [7855, [16, -6, 9, -14, 5, -10], [36, 2, -28, 10, -20], 0, -312, 480],
      [7859, [-6, 7, -4, -17, 16, 4], [-4, -2, -34, 32, 8], 0, 240, -144],
      [8027, [18, 7, -24, -15, -18, 32], [68, -66, -30, -36, 64], 0, 552, 432],
      [8770, [-16, 8, 18, 22, 16, -48], [-32, 52, 44, 32, -96], 0, -1032, 48],
      [8979, [4, -22, -22, 14, -18, 44], [-32, -48, 28, -36, 88], 0, 1104, -768],
      [9640, [10, -23, 7, 15, 13, -22], [-16, 4, 30, 26, -44], 0, -384, -96],
      [10032, [-16, -12, 20, -6, 2, 12], [-72, 56, -12, 4, 24], 0, 408, -720],
      [10374, [18, -5, 0, -13, -12, 12], [44, -18, -26, -24, 24], 0, 96, 432],
      [10452, [18, -5, 0, -9, 24, -28], [44, -18, -18, 48, -56], 0, -576, 528],
      [10540, [-4, -12, -7, 6, -1, 18], [-36, -10, 12, -2, 36], 0, 552, -576],
      [10901, [-24, -12, -14, 18, -16, 48], [-96, -4, 36, -32, 96], 0, 1320, -1392],
      [11462, [-10, -3, 14, 17, 14, -32], [-36, 38, 34, 28, -64], 0, -624, -144],
      [11554, [-14, 23, 12, -3, 14, -32], [4, 38, -6, 28, -64], 0, -744, 336],
      [11827, [-12, -20, 19, 6, 17, -10], [-76, 50, 12, 34, -20], 0, 24, -720],
      [12078, [-4, -6, -8, 14, -12, 16], [-24, -12, 28, -24, 32], 0, 384, -384],
      [12136, [-12, -10, 2, -24, 4, 40], [-56, 16, -48, 8, 80], 0, 1128, -864],
      [12516, [20, 22, 18, 14, 10, -84], [104, 16, 28, 20, -168], 0, -2280, 1776],
      [12867, [-24, 24, -2, 20, -22, 4], [-24, 20, 40, -44, 8], 0, -48, -144],
      [12868, [-6, 21, -23, 1, -19, 26], [24, -40, 2, -38, 52], 0, 480, 48],
      [13218, [-2, -13, 19, 9, 1, -14], [-32, 40, 18, 2, -28], 0, -312, -144],
      [13526, [18, 3, 20, 5, 10, -56], [60, 22, 10, 20, -112], 0, -1488, 1104],
      [15312, [-10, 17, -8, -23, -12, 36], [4, -6, -46, -24, 72], 0, 768, -144],
      [15900, [6, 7, 3, -23, 21, -14], [32, 0, -46, 42, -28], 0, -264, 384],
      [16994, [-4, 14, -6, 22, -14, -12], [16, -8, 44, -28, -24], 0, -432, 288],
      [17009, [-12, 8, -7, 20, -23, 14], [-20, -2, 40, -46, 28], 0, 216, -240],
      [17153, [-6, -3, 9, -3, 9, -6], [-24, 24, -6, 18, -12], 0, -48, -192],
      [17195, [14, 21, 8, -23, -16, -4], [84, 2, -46, -32, -8], 0, -480, 1104],
      [17306, [-12, 4, 7, 6, 5, -10], [-28, 26, 12, 10, -20], 0, -168, -192],
      [17840, [-10, 1, -10, 23, 8, -12], [-28, -10, 46, 16, -24], 0, -120, -336]
    ]
  },
  {
    "depth": 5,
    "odd_part": 3,
    "valid_source_units": 10,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [0, 3],
```

```
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3]
    ],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [0, 3], [1, 3], [2, 3],
[3, 3], [4, 3]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        367,
        [-20, -18, 24, 11, 22, -15, -10, 15, -9, -2, 11, -9],
        [-96, 68, 22, 44, -30, 0, -8, -4, 22, -18],
        0,
        -432,
        -2592
      ],
      [
        1218,
        [9, 3, -21, -13, 15, -17, 13, -9, -9, 3, -1, 27],
        [33, -51, -26, 30, -34, 21, -31, 6, -2, 54],
        0,
        2016,
        288
      ],
      [
        1587,
        [-12, -4, -24, -19, 21, -4, 6, -7, 1, -20, -4, 66],
        [-44, -36, -38, 42, -8, 4, -4, -40, -8, 132],
        0,
        5592,
        -3216
      ],
      [
        5168,
        [8, 9, 7, 11, -17, 14, -16, 21, 17, -15, -5, -34],
        [42, 6, 22, -34, 28, -6, 50, -30, -10, -68],
        0,
        -3264,
        2400
      ],
      [
        8003,
        [-21, -19, -21, 23, -24, 2, 23, 3, 2, -23, 11, 44],
        [-101, -21, 46, -48, 4, 75, -19, -46, 22, 88],
        0,
        4032,
        -4080
      ],
      [
        8762,
        [9, -8, -20, 3, 20, 8, 13, -22, 17, -23, -3, 6],
        [11, -49, 6, 40, 16, -5, 21, -46, -6, 12],
        0,
        504,
```

```
        -384
      ],
      [
        9206,
        [23, 1, -7, 6, 24, -21, -15, 16, -6, -7, -7, -7],
        [71, -37, 12, 48, -42, -13, 3, -14, -14, -14],
        0,
        -1440,
        2544
      ],
      [
        12376,
        [10, 24, -17, -3, 15, 17, 20, -15, -22, 24, -12, -41],
        [78, -44, -6, 30, 34, 30, -64, 48, -24, -82],
        0,
        -3480,
        2928
      ],
      [
        13315,
        [-19, -20, -14, 5, 16, 22, -17, 9, -10, -8, -10, 46],
        [-97, -9, 10, 32, 44, -33, -3, -16, -20, 92],
        0,
        4512,
        -4992
      ],
      [
        14823,
        [13, -23, 12, 6, -21, -3, 23, 16, 11, 1, -2, -33],
        [-7, 11, 12, -42, -6, 101, -1, 2, -4, -66],
        0,
        -3312,
        1344
      ]
    ]
  },
  {
    "depth": 5,
    "odd_part": 5,
    "valid_source_units": 31,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5]
    ],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [0, 5], [1, 5], [2, 5],
[3, 5], [4, 5]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
```

```
  "target_infinity_sum", "target_reciprocal_sum"
],
"accepted_cases": [
  [
    1,
    [3, -18, 2, -6, -11, -10, 1, 20, 15, 20, 24, -40],
    [-27, 1, -12, -22, -20, 43, 29, 40, 48, -80],
    0,
    -3744,
    -1584
  ],
  [
    751,
    [18, -7, -4, 13, -1, 3, -6, -19, -22, 1, 19, 5],
    [40, -26, 26, -2, 6, -56, -38, 2, 38, 10],
    0,
    1872,
    1584
  ],
  [
    815,
    [6, 16, -18, 15, -3, 8, 20, -5, -12, -18, 19, -28],
    [50, -42, 30, -6, 16, 50, -44, -36, 38, -56],
    0,
    -3576,
    3264
  ],
  [
    1185,
    [23, -7, 18, -16, 24, -6, 19, -24, 24, 11, -20, -46],
    [55, 13, -32, 48, -12, 9, 29, 22, -40, -92],
    0,
    -8040,
    4992
  ],
  [
    1985,
    [-24, -10, 2, 3, 8, -7, 6, 5, -15, -18, -19, 69],
    [-92, 28, 6, 16, -14, 28, -36, -36, -38, 138],
    0,
    8472,
    -5952
  ],
  [
    2149,
    [-1, 1, 17, -13, -18, 24, -7, -1, 10, -21, 3, 6],
    [-1, 35, -26, -36, 48, -23, 27, -42, 6, 12],
    0,
    960,
    384
  ],
  [
    5625,
    [-14, -8, 13, -20, 11, 0, 22, 8, 15, 2, 15, -44],
    [-58, 40, -40, 22, 0, 82, 8, 4, 30, -88],
    0,
    -5232,
    -2256
  ],
```

```
[
  6024,
  [24, 17, 13, 2, 19, 19, 6, 4, 11, 21, 23, -159],
  [106, 2, 4, 38, 38, 26, 16, 42, 46, -318],
  0,
  -21432,
  9696
],
[
  6577,
  [-15, 2, -14, -23, 17, -19, 23, -6, -7, -17, -6, 65],
  [-41, -13, -46, 34, -38, 57, -37, -34, -12, 130],
  0,
  8568,
  -3984
],
[
  6776,
  [23, 10, 9, -22, 14, 12, -17, 7, 4, -15, -11, -14],
  [89, -5, -44, 28, 24, -37, 25, -30, -22, -28],
  0,
  -3144,
  5856
],
[
  6991,
  [9, -11, 17, -18, 18, -23, -17, 19, 22, -12, 11, -15],
  [5, 25, -36, 36, -46, -13, 61, -24, 22, -30],
  0,
  -1992,
  1008
],
[
  7232,
  [-15, 23, -18, 8, -9, 23, 3, 3, -7, 10, 4, -25],
  [1, -21, 16, -18, 46, 15, -17, 20, 8, -50],
  0,
  -2760,
  -240
],
[
  8222,
  [11, -15, -6, -24, -16, -6, -21, -18, -18, -19, 6, 126],
  [3, -23, -48, -32, -12, -99, -15, -38, 12, 252],
  0,
  18552,
  -3600
],
[
  8794,
  [0, 12, -20, -8, 3, 5, -8, -16, -22, -10, 9, 55],
  [24, -40, -16, 6, 10, -56, -36, -20, 18, 110],
  0,
  8568,
  -1008
],
[
  10404,
  [-12, 2, -10, 12, -19, 15, -16, -14, -1, 10, -6, 39],
```

```
  [-32, -8, 24, -38, 30, -76, 14, 20, -12, 78],
  0,
  6144,
  -3600
],
[
  11071,
  [22, -23, 20, -24, -10, -17, 20, -20, -1, -19, 9, 43],
  [20, 18, -48, -20, -34, 20, -22, -38, 18, 86],
  0,
  5880,
  1104
],
[
  11934,
  [-9, -4, -6, 8, 18, -23, 19, 15, -13, 5, -11, 1],
  [-35, -3, 16, 36, -46, 87, -45, 10, -22, 2],
  0,
  -960,
  -1440
],
[
  11989,
  [23, 13, 23, -5, 8, -2, 1, 3, 4, -3, -5, -60],
  [95, 23, -10, 16, -4, 9, 7, -6, -10, -120],
  0,
  -9840,
  8496
],
[
  12409,
  [3, 2, 22, 8, 6, -23, 1, -4, -1, -24, -11, 21],
  [13, 41, 16, 12, -46, -5, -3, -48, -22, 42],
  0,
  984,
  2592
],
[
  12788,
  [22, 15, 1, -19, -20, -1, -24, 22, -9, -22, 14, 21],
  [96, -20, -38, -40, -2, -28, 6, -44, 28, 42],
  0,
  3072,
  5232
],
[
  13865,
  [23, -12, 9, 14, -13, 7, 9, 24, -24, 16, 12, -65],
  [45, -5, 28, -26, 14, 75, -57, 32, 24, -130],
  0,
  -8832,
  4560
],
[
  14677,
  [23, -4, 1, -3, 2, -5, -7, 18, 3, 15, 16, -59],
  [61, -21, -6, 4, -10, 15, 13, 30, 32, -118],
  0,
  -7488,
```

```
    4320
  ],
  [
    16029,
    [-5, -6, -23, 21, -7, -12, -13, 23, -12, -14, -2, 50],
    [-27, -41, 42, -14, -24, 7, -11, -28, -4, 100],
    0,
    6768,
    -3216
  ],
  [
    16048,
    [5, 14, -24, 5, -1, 7, 21, -17, -16, 14, 11, -19],
    [43, -53, 10, -2, 14, 29, -53, 28, 22, -38],
    0,
    -1800,
    1728
  ],
  [
    16354,
    [-1, -17, 21, -24, -13, 6, 7, -10, -18, 17, 16, 16],
    [-37, 43, -48, -26, 12, 1, -43, 34, 32, 32],
    0,
    3936,
    -2496
  ],
  [
    16656,
    [4, -19, 6, -1, 17, 17, -4, 7, -12, -1, 7, -21],
    [-26, 8, -2, 34, 34, 2, -20, -2, 14, -42],
    0,
    -2232,
    -1440
  ],
  [
    16769,
    [18, 24, -13, -7, 15, 3, 16, -21, 3, 4, -15, -27],
    [102, -44, -14, 30, 6, 6, -10, 8, -30, -54],
    0,
    -5136,
    6384
  ],
  [
    16904,
    [-17, 16, -7, 21, -19, 20, -17, 21, 4, 18, -2, -38],
    [-19, 3, 42, -38, 40, -9, 25, 36, -4, -76],
    0,
    -4824,
    -624
  ],
  [
    17087,
    [1, -13, 1, 0, -14, 21, 19, -13, -20, -6, 11, 13],
    [-23, 1, 0, -28, 42, 31, -59, -12, 22, 26],
    0,
    2712,
    -1824
  ],
  [
```

```
        17470,
        [3, -17, -17, -15, 9, -11, -19, -5, -3, -17, -7, 99],
        [-25, -37, -30, 18, -22, -67, 13, -34, -14, 198],
        0,
        13968,
        -4944
      ],
      [
        17934,
        [-17, 16, 13, -4, -14, -6, -7, -2, -3, 10, -20, 34],
        [-19, 43, -8, -28, -12, -25, 1, 20, -40, 68],
        0,
        3744,
        -624
      ]
    ]
  },
  {
    "depth": 5,
    "odd_part": 9,
    "valid_source_units": 15,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9],
      [4, 9],
      [5, 9]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9],
      [4, 9]
    ],
```

```
"columns": [
  "trial", "source_exponents", "target_exponents", "target_weight_sum",
  "target_infinity_sum", "target_reciprocal_sum"
],
"accepted_cases": [
  [
    107,
    [1, -1, -14, 8, 3, 19, -17, 3, -14, 5, 21, 6, -18, -20, -9, 11, -1, 17],
    [1, -29, 16, 6, 38, -45, -11, 10, 42, 12, -94, 0, 22, -2, 34],
    0,
    6864,
    -4320
  ],
  [
    1381,
    [4, -11, 3, 16, 18, 20, 8, -11, 24, 16, -5, -8, 20, -22, 2, 20, -20, -74],
    [-10, 2, 32, 36, 40, 2, 40, 32, -10, -16, 16, -16, 40, -40, -148],
    0,
    -22224,
    2256
  ],
  [
    3033,
    [-9, 12, 9, -18, -18, -6, -3, -17, 21, 15, -8, 22, 22, 3, 11, 5, 9, -50],
    [-3, 27, -36, -36, -12, -43, 45, 30, -16, 44, 72, 0, 10, 18, -100],
    0,
    -10440,
    0
  ],
  [
    3295,
    [23, 0, 2, -11, -13, 14, 12, 4, -24, -2, 19, -17, 15, 23, 14, 2, 14, -75],
    [69, -19, -22, -26, 28, 44, -60, -4, 38, -34, 91, 13, 4, 28, -150],
    0,
    -19200,
    9792
  ],
  [
    3514,
    [10, 0, 12, -3, 23, -12, 17, -17, 20, -1, -12, 3, 1, 22, 1, 3, -4, -63],
    [30, 14, -6, 46, -24, 17, 23, -2, -24, 6, 47, 1, 6, -8, -126],
    0,
    -18168,
    7584
  ],
  [
    3558,
    [-3, 16, 0, 18, 19, 9, -18, -11, -7, -8, 5, -15, 17, -21, 16, 24, -7, -34],
    [23, 3, 36, 38, 18, -76, 4, -16, 10, -30, 9, 15, 48, -14, -68],
    0,
    -9552,
    2256
  ],
  [
    5952,
    [-14, -5, 4, -20, -11, 7, -13, 24, -6, 4, -8, 15, 7, -8, 14, -15, -11, 36],
    [-52, 22, -40, -22, 14, 9, 1, 8, -16, 30, 5, 21, -30, -22, 72],
    0,
    9192,
```

```
  -6912
],
[
  6488,
  [-24, -19, 0, -19, -21, -4, -21, 4, 22, 1, 0, 0, 9, 22, -2, 7, -19, 64],
  [-110, 24, -38, -42, -8, -55, 65, 2, 0, 0, 71, -13, 14, -38, 128],
  0,
  16176,
  -16224
],
[
  6764,
  [-11, -24, 6, 9, 20, 3, 20, 19, 13, -7, -4, -1, 9, -10, 17, -13, -22, -24],
  [-81, 23, 18, 40, 6, 98, 6, -14, -8, -2, 7, 25, -26, -44, -48],
  0,
  -10176,
  -3888
],
[
  6807,
  [21, 9, 0, 13, 15, -1, -14, -14, 12, -15, -10, -19, 23, 24, 12, -23, -11, -22],
  [81, -21, 26, 30, -2, -70, 38, -30, -20, -38, 117, 1, -46, -22, -44],
  0,
  -10800,
  10176
],
[
  10428,
  [-15, 24, 4, -22, -22, -23, 17, 21, -1, -2, 0, -9, 18, -8, -14, 7, -23, 48],
  [3, 23, -44, -44, -46, 93, -19, -4, 0, -18, 38, -46, 14, -46, 96],
  0,
  8568,
  3504
],
[
  10502,
  [-12, 4, -7, -11, 20, -13, 2, -1, -13, -4, 2, -20, 12, -20, -16, -18, 2, 93],
  [-28, -2, -22, 40, -26, 4, -28, -8, 4, -40, -4, -44, -36, 4, 186],
  0,
  22656,
  -5520
],
[
  11886,
  [-17, -20, 2, -2, -19, -9, 12, 12, -10, -1, -18, -9, -17, 21, -3, 12, -8, 74],
  [-91, 21, -4, -38, -18, 60, -32, -2, -36, -18, -9, 11, 24, -16, 148],
  0,
  18720,
  -10608
],
[
  14915,
  [6, 1, -1, -4, 18, -16, -23, 14, -1, -24, -23, -19, -23, -2, -21, 3, 8, 107],
  [20, -8, -8, 36, -32, -41, 21, -48, -46, -38, -73, -19, 6, 16, 214],
  0,
  27432,
  -1104
],
[
```

```
        15440,
        [-17, 9, 6, 19, -15, -3, -3, 8, -4, 2, 3, -18, 20, -12, 11, 4, 19, -29],
        [-33, 29, 38, -30, -6, 7, -5, 4, 6, -36, 36, 2, 8, 38, -58],
        0,
        -6672,
        -1056
      ]
    ]
  },
  {
    "depth": 5,
    "odd_part": 15,
    "valid_source_units": 2,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5],
      [0, 15],
      [1, 15],
      [2, 15],
      [3, 15],
      [4, 15],
      [5, 15]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [0, 15],
      [1, 15],
      [2, 15],
```

```
      [3, 15],
      [4, 15]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        4153,
        [
          -24, -15, -3, 16, 18, -2, -2, 20, 21, 21, -10, -6, -20, -11, 24, 22, -15, 4,
          8, 5, 18, -2, -7, -60
        ],
        [
          -102, 18, 32, 36, -4, 34, 44, 42, -20, -12, -82, 68, 44, -30, 8, 34, 28, -4,
          -14, -120
        ],
        0,
        -28680,
        -15648
      ],
      [
        14417,
        [
          18, -11, -4, -2, 1, 12, 14, -14, -18, -20, -20, -6, 20, -17, 18, 24, -21,
          -24, 16, -8, -7, 0, -15, 64
        ],
        [
          32, -26, -4, 2, 24, 14, -50, -40, -40, -12, 26, 16, 48, -42, -48, 32, -30,
          0, -30, 128
        ],
        0,
        20520,
        4752
      ]
    ]
  },
  {
    "depth": 6,
    "odd_part": 1,
    "valid_source_units": 52,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [6, 1]],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        335,
        [-16, -10, 11, -20, 13, -18, 40],
        [-68, 38, -40, 26, -36, 80],
        0,
        2040,
        -1776
      ],
      [
```

```
  727,
  [-8, -24, 24, -8, 20, 20, -24],
  [-72, 56, -16, 40, 40, -48],
  0,
  -600,
  -1344
],
[1809, [-2, -13, -9, -20, 3, 3, 38], [-32, -16, -40, 6, 6, 76], 0, 2352, -1488],
[
  2085,
  [20, 20, 11, -13, -9, -7, -22],
  [100, 2, -26, -18, -14, -44],
  0,
  -1776,
  2880
],
[2576, [2, -9, 10, 10, 4, -7, -10], [-12, 18, 20, 8, -14, -20], 0, -696, 48],
[
  3256,
  [18, 13, -21, 4, 13, -17, -10],
  [80, -60, 8, 26, -34, -20],
  0,
  -984,
  1680
],
[
  3576,
  [-10, -17, 9, -17, 7, 16, 12],
  [-64, 28, -34, 14, 32, 24],
  0,
  1248,
  -1728
],
[
  3674,
  [-8, -6, -13, -11, 11, -7, 34],
  [-36, -18, -22, 22, -14, 68],
  0,
  1968,
  -1488
],
[
  3854,
  [-8, -20, 16, 9, 22, -13, -6],
  [-64, 40, 18, 44, -26, -12],
  0,
  -360,
  -1152
],
[3874, [2, -19, -7, -11, 21, 6, 8], [-32, -16, -22, 42, 12, 16], 0, 888, -1248],
[
  3889,
  [20, 22, 14, 8, -22, -6, -36],
  [104, 8, 16, -44, -12, -72],
  0,
  -2664,
  3312
],
[
```

```
  4284,
  [16, -24, -12, -12, 12, -12, 32],
  [0, -40, -24, 24, -24, 64],
  0,
  1680,
  -720
],
[
  4412,
  [-12, 18, 24, 21, 12, -9, -54],
  [0, 60, 42, 24, -18, -108],
  0,
  -3264,
  1248
],
[
  4743,
  [12, -20, -5, 20, -17, -6, 16],
  [-4, -22, 40, -34, -12, 32],
  0,
  672,
  -288
],
[
  5869,
  [20, 2, -6, 12, 14, -14, -28],
  [64, -32, 24, 28, -28, -56],
  0,
  -1920,
  1728
],
[5969, [-16, 0, 20, 4, 14, -10, -12], [-48, 56, 8, 28, -20, -24], 0, -768, -528],
[
  6044,
  [-22, 1, -17, 8, -17, 21, 26],
  [-64, -12, 16, -34, 42, 52],
  0,
  2040,
  -2112
],
[6127, [2, -3, -3, -11, 5, -22, 32], [0, -8, -22, 10, -44, 64], 0, 1320, -288],
[6755, [-12, -6, -2, -6, 12, 18, -4], [-48, 8, -12, 24, 36, -8], 0, 432, -1344],
[7365, [10, -15, 7, 9, 13, -12, -12], [0, 4, 18, 26, -24, -24], 0, -864, 240],
[7553, [-18, -19, 10, 5, 22, 8, -8], [-92, 38, 10, 44, 16, -16], 0, 120, -2064],
[
  8159,
  [4, -2, -12, 10, -14, -14, 28],
  [8, -28, 20, -28, -28, 56],
  0,
  1152,
  -144
],
[10036, [6, 23, 17, 13, 9, 24, -92], [64, 28, 26, 18, 48, -184], 0, -4752, 2688],
[
  10430,
  [-24, -14, 19, 12, 15, 20, -28],
  [-100, 62, 24, 30, 40, -56],
  0,
  -792,
```

```
  -1872
],
[
  10690,
  [22, 13, -12, -20, -4, -17, 18],
  [92, -46, -40, -8, -34, 36],
  0,
  384,
  1824
],
[
  10696,
  [24, -24, 18, 19, -18, -13, -6],
  [24, 12, 38, -36, -26, -12],
  0,
  -888,
  1056
],
[
  11165,
  [16, -12, 24, 20, 4, -20, -32],
  [24, 32, 40, 8, -40, -64],
  0,
  -2376,
  1488
],
[
  11520,
  [-18, 23, 13, 19, 13, 6, -56],
  [-8, 44, 38, 26, 12, -112],
  0,
  -2952,
  768
],
[
  11830,
  [22, -3, 10, 0, 12, -15, -26],
  [60, -2, 0, 24, -30, -52],
  0,
  -1896,
  1872
],
[
  12369,
  [12, 14, -2, 12, -18, -14, -4],
  [64, -16, 24, -36, -28, -8],
  0,
  -864,
  1776
],
[12489, [-20, 2, 24, 12, -22, 4, 0], [-56, 68, 24, -44, 8, 0], 0, -48, -672],
[
  12585,
  [-6, 15, -16, 24, -24, -15, 22],
  [12, -26, 48, -48, -30, 44],
  0,
  696,
  144
],
```

```
[
  12896,
  [20, -24, -15, 12, -15, -18, 40],
  [12, -50, 24, -30, -36, 80],
  0,
  1752,
  -336
],
[
  13015,
  [-8, -12, -22, 12, 2, -12, 40],
  [-48, -36, 24, 4, -24, 80],
  0,
  2184,
  -1872
],
[13291, [20, 22, -22, -18, 0, 2, -4], [104, -64, -36, 0, 4, -8], 0, -360, 2016],
[
  13293,
  [12, -12, -5, -22, -17, -8, 52],
  [12, -22, -44, -34, -16, 104],
  0,
  2592,
  -384
],
[
  13696,
  [2, -19, 7, -18, -19, 13, 34],
  [-32, 12, -36, -38, 26, 68],
  0,
  2136,
  -1152
],
[13929, [18, 5, -5, -21, 9, -14, 8], [64, -28, -42, 18, -28, 16], 0, 48, 1296],
[
  13992,
  [18, -21, 16, -14, -16, 23, -6],
  [12, 14, -28, -32, 46, -12],
  0,
  24,
  336
],
[14516, [6, 13, 0, 15, 12, 14, -60], [44, -6, 30, 24, 28, -120], 0, -3048, 1584],
[
  14901,
  [6, -15, -6, -14, 12, 23, -6],
  [-12, -18, -28, 24, 46, -12],
  0,
  384,
  -720
],
[15863, [2, 21, -15, -8, -3, -7, 10], [48, -32, -16, -6, -14, 20], 0, 288, 864],
[
  15902,
  [6, 13, 10, 9, 20, -22, -36],
  [44, 14, 18, 40, -44, -72],
  0,
  -2544,
  1776
```

```
      ],
      [
        16060,
        [12, 18, 14, 18, 18, 24, -104],
        [72, 16, 36, 36, 48, -208],
        0,
        -5352,
        2880
      ],
      [
        16226,
        [10, -21, -24, 24, -12, -15, 38],
        [-12, -58, 48, -24, -30, 76],
        0,
        1824,
        -1008
      ],
      [16332, [-14, 3, -20, 5, 12, 10, 4], [-36, -26, 10, 24, 20, 8], 0, 720, -1344],
      [16356, [14, 3, 11, -21, -11, 0, 4], [48, 8, -42, -22, 0, 8], 0, -24, 1248],
      [
        16405,
        [22, -3, -16, -23, -22, 14, 28],
        [60, -54, -46, -44, 28, 56],
        0,
        1656,
        624
      ],
      [
        16425,
        [-10, -13, 13, 24, 11, -11, -14],
        [-56, 36, 48, 22, -22, -28],
        0,
        -864,
        -816
      ],
      [
        17354,
        [18, -1, 24, -4, -8, -19, -10],
        [52, 30, -8, -16, -38, -20],
        0,
        -1296,
        1920
      ],
      [17355, [6, 1, -14, -24, -2, 23, 10], [20, -34, -48, -4, 46, 20], 0, 1104, -192],
      [17938, [10, 23, -24, -5, -16, 12, 0], [76, -58, -10, -32, 24, 0], 0, 48, 1344]
    ]
  },
  {
    "depth": 6,
    "odd_part": 3,
    "valid_source_units": 8,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
```

```
      [6, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [6, 3]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        4776,
        [-3, 22, -16, -24, -10, -16, 5, 11, 5, 8, 4, 4, 15, -5],
        [35, -29, -48, -20, -32, 10, 43, 5, 8, 8, 30, -10],
        0,
        360,
        2016
      ],
      [
        6979,
        [-23, -3, 24, -8, 2, 4, 14, 23, 16, 20, 14, -12, 13, -84],
        [-75, 71, -16, 4, 8, 28, 101, 17, 28, -24, 26, -168],
        0,
        -13656,
        -480
      ],
      [
        12613,
        [-5, 7, -12, -11, 9, 23, 3, -17, -9, -24, 1, -9, 1, 43],
        [-1, -19, -22, 18, 46, 6, -69, -31, 2, -18, 2, 86],
        0,
        8496,
        -3696
      ],
      [
        13829,
        [9, 8, -5, 15, -20, -13, 14, 3, -21, 12, -21, 7, 20, -8],
        [43, -19, 30, -40, -26, 28, -33, 21, -42, 14, 40, -16],
        0,
        528,
        2448
```

```
      ],
      [
        14983,
        [-10, -10, 5, -22, -6, -12, -3, 24, 11, -24, 16, 15, -19, 35],
        [-50, 20, -44, -12, -24, -6, 94, -72, 32, 30, -38, 70],
        0,
        4992,
        -2976
      ],
      [
        17056,
        [-17, 17, 0, -8, -20, 13, 17, -7, 24, -3, 7, 15, 1, -39],
        [-17, 17, -16, -40, 26, 34, 27, 1, 14, 30, 2, -78],
        0,
        -5280,
        -384
      ],
      [
        17739,
        [1, -19, 8, -20, 3, 23, -10, -17, 20, 10, -12, -19, 20, 12],
        [-35, 15, -40, 6, 46, -20, -11, 37, -24, -38, 40, 24],
        0,
        3192,
        -3312
      ],
      [
        17908,
        [0, 13, -18, 22, -8, 12, -17, -8, -17, 13, 13, -1, -13, 9],
        [26, -36, 44, -16, 24, -34, -58, 34, 26, -2, -26, 18],
        0,
        72,
        528
      ]
    ]
  },
  {
    "depth": 6,
    "odd_part": 5,
    "valid_source_units": 19,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5],
      [6, 5]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
```

```
    [2, 1],
    [3, 1],
    [4, 1],
    [5, 1],
    [0, 5],
    [1, 5],
    [2, 5],
    [3, 5],
    [4, 5],
    [5, 5]
  ],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [
      684,
      [9, -4, -20, -21, -5, -22, -19, 11, 22, -15, 6, 2, 7, 49],
      [19, -49, -42, -10, -44, -38, 77, -41, 12, 4, 14, 98],
      0,
      14928,
      -1344
    ],
    [
      1340,
      [6, 10, 12, 15, -1, -11, -17, -18, -4, 18, -12, -1, 16, -13],
      [38, 18, 30, -2, -22, -34, -62, 54, -24, -2, 32, -26],
      0,
      -3192,
      7008
    ],
    [
      2424,
      [21, 5, -17, 18, -9, -21, -21, -17, 13, -6, 5, -6, 20, 15],
      [73, -55, 36, -18, -42, -42, -25, 5, 10, -12, 40, 30],
      0,
      5592,
      7152
    ],
    [
      3569,
      [20, 3, -7, 8, -18, 16, 8, 10, -14, 19, 10, 14, 13, -82],
      [66, -34, 16, -36, 32, 16, 2, 28, 20, 28, 26, -164],
      0,
      -21552,
      8832
    ],
    [
      5464,
      [21, 21, -10, 10, 8, -23, -5, 15, 3, -18, -22, 20, -13, -7],
      [105, -41, 20, 16, -46, -10, 51, -51, -44, 40, -26, -14],
      0,
      -4680,
      14688
    ],
    [
      5713,
      [-20, 0, 18, -20, -6, -7, 21, -10, -15, -11, 20, 3, -18, 45],
```

```
  [-60, 56, -40, -12, -14, 42, -60, -12, 40, 6, -36, 90],
  0,
  13056,
  -8640
],
[
  7345,
  [-21, 22, 19, -7, -6, 20, -19, 23, 13, -2, 24, 21, 2, -89],
  [-19, 59, -14, -12, 40, -38, 95, -27, 48, 42, 4, -178],
  0,
  -25944,
  4080
],
[
  7372,
  [-16, -12, -5, -3, 21, 6, -23, 2, 7, -1, 4, -9, -6, 35],
  [-72, 6, -6, 42, 12, -46, 20, -4, 8, -18, -12, 70],
  0,
  8712,
  -9936
],
[
  8603,
  [-16, -7, 13, -21, -5, 20, -20, -16, 13, -4, -21, 8, 24, 32],
  [-62, 42, -42, -10, 40, -40, -22, 8, -42, 16, 48, 64],
  0,
  12984,
  -8928
],
[
  9761,
  [-19, -8, -4, -11, -3, -18, 3, 23, 12, 12, 0, 19, -13, 7],
  [-73, 11, -22, -6, -36, 6, 93, 1, 0, 38, -26, 14],
  0,
  1584,
  -9024
],
[
  10281,
  [-14, -15, 13, -18, 19, -6, -13, -6, 23, 10, -6, -16, -4, 33],
  [-72, 40, -36, 38, -12, -26, 28, 26, -12, -32, -8, 66],
  0,
  7944,
  -8112
],
[
  10315,
  [-4, 17, 22, -7, -14, -13, -5, -12, -7, 17, 3, -15, -19, 37],
  [22, 48, -14, -28, -26, -10, -50, 46, 6, -30, -38, 74],
  0,
  7032,
  4992
],
[
  11732,
  [7, 10, -5, 13, 2, 13, -24, -21, 7, 9, -20, -24, -3, 36],
  [41, -17, 26, 4, 26, -48, -49, 39, -40, -48, -6, 72],
  0,
  7488,
```

```
        4944
      ],
      [
        12245,
        [-8, 15, 10, 22, 18, -13, -10, -4, -17, -10, 8, -16, 23, -18],
        [6, 28, 44, 36, -26, -20, -46, -16, 16, -32, 46, -36],
        0,
        -3960,
        3648
      ],
      [
        12758,
        [-18, 23, -20, 16, -13, -5, -13, -22, 11, -23, -15, 16, 8, 55],
        [-8, -22, 32, -26, -10, -26, -44, -24, -30, 32, 16, 110],
        0,
        17976,
        -4272
      ],
      [
        13668,
        [15, -10, 11, -1, 20, 9, 20, -19, 12, 10, 10, 15, 2, -94],
        [25, 7, -2, 40, 18, 40, -33, 39, 20, 30, 4, -188],
        0,
        -26016,
        5328
      ],
      [
        14060,
        [1, -15, 4, -6, -16, 18, 22, 3, -1, -24, 18, -4, 14, -14],
        [-27, 7, -12, -32, 36, 44, 7, -51, 36, -8, 28, -28],
        0,
        -648,
        -4608
      ],
      [
        15916,
        [22, -23, -16, 8, -23, 3, 9, -4, -12, 1, -6, -14, 2, 53],
        [20, -54, 16, -46, 6, 18, -36, 6, -12, -28, 4, 106],
        0,
        16080,
        -2400
      ],
      [
        17635,
        [-4, -12, 12, 13, -1, -21, -15, -6, -21, 1, 20, 22, 21, -9],
        [-36, 28, 26, -2, -42, -30, -60, 8, 40, 44, 42, -18],
        0,
        1296,
        -4320
      ]
    ]
  },
  {
    "depth": 6,
    "odd_part": 9,
    "valid_source_units": 8,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
```

```
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [6, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9],
      [4, 9],
      [5, 9],
      [6, 9]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9],
      [4, 9],
      [5, 9]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        3252,
        [
          3, 1, 16, -12, -2, -14, 3, -6, 9, 8, -10, -23, -13, 9, 3, 4, -9, 4, 12, -9,
          26
        ],
        [11, 29, -24, -4, -28, 6, 0, 22, -20, -46, -26, 18, 17, -21, 8, 24, -18, 52],
        0,
        13128,
        5328
      ],
      [
        3409,
```

```
    [
      8, 7, 7, 20, 7, 24, 11, 16, -18, 18, 4, 12, -23, -3, -22, 8, -12, -19, 8,
      17, -70
    ],
    [38, 6, 40, 14, 48, 22, 12, 20, 8, 24, -46, -6, -50, -2, -38, 16, 34, -140],
    0,
    -36288,
    16608
  ],
  [
    5512,
    [
      7, 1, 2, -24, 11, 16, 3, 19, -24, -1, 20, -20, 17, 17, -4, 1, 5, -12, 23,
      -11, -46
    ],
    [23, -3, -48, 22, 32, 6, 9, -21, 40, -40, 34, 34, -10, 14, -24, 46, -22, -92],
    0,
    -22032,
    4560
  ],
  [
    5763,
    [
      7, -24, 2, 20, -18, -23, 9, -7, 14, -12, -7, 11, 10, -3, -22, -10, -5, -6,
      -14, 16, 62
    ],
    [
      -27, -3, 40, -36, -46, 18, 7, -17, -14, 22, 20, -6, -86, 12, -12, -28, 32,
      124
    ],
    0,
    37656,
    -9984
  ],
  [
    8612,
    [
      13, -23, 1, -14, -4, 19, 20, -9, -12, -14, -9, 17, -19, 14, -12, 16, 17, -3,
      1, 21, -20
    ],
    [-7, -11, -28, -8, 38, 40, -51, -19, -18, 34, -38, 28, -4, 46, -6, 2, 42, -40],
    0,
    -1872,
    -10224
  ],
  [
    11440,
    [
      14, 19, -5, -13, 13, 5, -17, 9, -2, 20, -6, 3, -13, -17, 5, 20, -1, 17, -8,
      -21, -22
    ],
    [
      80, -24, -26, 26, 10, -34, 23, 31, -12, 6, -26, -34, 55, -7, 34, -16, -42,
      -44
    ],
    0,
    -23328,
    23472
  ],
```

```
      [
        17778,
        [
          21, 19, 0, -12, 10, -22, 14, 24, 9, 0, -16, 1, 22, 12, 5, -19, 24, 21, -7,
          8, -114
        ],
        [
          101, -21, -24, 20, -44, 28, 90, -24, -32, 2, 44, 24, -23, 43, 42, -14, 16,
          -228
        ],
        0,
        -57768,
        31632
      ],
      [
        17886,
        [
          -17, -14, -2, 2, 3, 3, 1, -8, 21, -13, -12, 10, -12, -18, 3, 8, 0, -16, -22,
          -24, 107
        ],
        [-79, 13, 4, 6, 6, 2, 18, -18, -24, 20, -24, -36, 25, -3, -32, -44, -48, 214],
        0,
        46272,
        -19536
      ]
    ]
  },
  {
    "depth": 6,
    "odd_part": 15,
    "valid_source_units": 9,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [6, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5],
      [6, 5],
      [0, 15],
      [1, 15],
      [2, 15],
      [3, 15],
      [4, 15],
```

```
    [5, 15],
    [6, 15]
  ],
  "target_coordinates": [
    [0, 1],
    [1, 1],
    [2, 1],
    [3, 1],
    [4, 1],
    [5, 1],
    [0, 3],
    [1, 3],
    [2, 3],
    [3, 3],
    [4, 3],
    [5, 3],
    [0, 5],
    [1, 5],
    [2, 5],
    [3, 5],
    [4, 5],
    [5, 5],
    [0, 15],
    [1, 15],
    [2, 15],
    [3, 15],
    [4, 15],
    [5, 15]
  ],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [
      283,
      [
        -5, 19, -15, 20, -16, -2, -19, 8, -24, -18, 5, -10, 0, 5, 3, -23, -23, -20,
        -11, -20, 2, 16, -7, 24, 21, 9, -7, 88
      ],
      [
        23, -25, 40, -32, -4, -38, -24, -44, 10, -20, 0, 10, -37, -49, -40, -22,
        -40, 4, 34, 32, 42, 18, -14, 176
      ],
      0,
      81216,
      -5280
    ],
    [
      331,
      [
        15, -6, -6, 18, 24, -9, 3, 18, 0, -16, -4, 12, 10, -1, -9, -5, -16, 18, 5,
        -16, -4, -12, 23, 2, 9, 11, -18, -46
      ],
      [
        33, -27, 36, 48, -18, 6, 54, -50, -8, 24, 20, -2, -37, -23, 36, 10, -32, -8,
        10, 16, 18, 22, -36, -92
      ],
      0,
```

```
    -50064,
    17856
  ],
  [
    5147,
    [
      -3, 11, 18, 6, 3, -20, 10, -14, 1, -17, -24, 14, -2, -3, 20, -11, -4, -17,
      16, 16, -13, -17, -5, -7, -5, 1, 14, 32
    ],
    [
      13, 39, 12, 6, -40, 20, -40, -20, -48, 28, -4, -6, 38, -28, -34, 32, 32,
      -26, -61, 3, -10, 2, 28, 64
    ],
    0,
    34440,
    7200
  ],
  [
    5308,
    [
      11, 3, -13, 0, 5, -20, -23, 15, 3, 19, -12, 23, 5, -24, 18, 15, 2, 8, 12,
      18, -24, -4, -14, 2, -22, -22, -24, 43
    ],
    [
      39, -37, 0, 10, -40, -46, 51, 23, -24, 46, 10, -48, 84, -14, 16, 24, 36,
      -48, -40, 8, -44, -44, -48, 86
    ],
    0,
    13152,
    24816
  ],
  [
    7265,
    [
      11, 13, 0, -5, 10, 3, -12, -10, 8, -2, -14, -8, 21, 11, 4, -19, 23, -13, 12,
      11, -2, 13, -20, -1, 15, 4, 2, -55
    ],
    [
      59, -11, -10, 20, 6, -24, -14, 6, -28, -16, 42, 22, -26, 42, -26, 24, 22,
      -4, -1, -15, 30, 8, 4, -110
    ],
    0,
    -44808,
    22032
  ],
  [
    9229,
    [
      21, 13, -15, 4, 15, -4, -16, -19, 22, 0, -19, -2, -16, -24, -5, -17, 5, -5,
      19, 15, 14, -9, -10, 19, -20, -13, -7, 54
    ],
    [
      89, -51, 8, 30, -8, -32, -13, 19, -38, -4, -32, -48, -49, 15, -10, 38, 30,
      28, -47, 47, -40, -26, -14, 108
    ],
    0,
    44256,
    25200
  ],
```

```
      [
        10237,
        [
          -9, -24, 24, -21, 17, 4, 9, 17, -9, -13, 15, -23, 10, -19, 16, 23, -5, -9,
          -24, -12, -11, -22, 0, -20, 11, 0, 2, 73
        ],
        [
          -75, 57, -42, 34, 8, 18, 33, -43, 30, -46, 20, -38, 94, -26, -18, -48, -24,
          -22, -66, -18, 22, 0, 4, 146
        ],
        0,
        60576,
        -18192
      ],
      [
        11745,
        [
          -5, -13, -4, 13, -14, 13, -11, 6, -23, -3, 15, 7, 19, -14, 8, -22, 2, -9,
          -2, -24, 22, -3, 8, 9, 15, -13, -2, 25
        ],
        [
          -41, -3, 26, -28, 26, -22, -28, -12, 30, 14, 38, -28, -20, -4, -18, -4, -48,
          44, 7, 21, 30, -26, -4, 50
        ],
        0,
        24216,
        -24240
      ],
      [
        13279,
        [
          -1, 15, -19, 6, -1, -11, -19, -16, -23, 3, 17, 12, -9, 12, 0, -10, 15, 15,
          -15, 24, -3, 19, -16, 20, 17, 9, 11, -52
        ],
        [
          27, -37, 12, -2, -22, -38, -94, 22, 34, 24, -18, 24, -20, 30, 30, -30, 48,
          -6, 25, 21, 34, 18, 22, -104
        ],
        0,
        -36264,
        -5760
      ]
    ]
  },
  {
    "depth": 7,
    "odd_part": 1,
    "valid_source_units": 38,
    "trials": 18000,
    "source_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [6, 1], [7, 1]],
    "target_coordinates": [[0, 1], [1, 1], [2, 1], [3, 1], [4, 1], [5, 1], [6, 1]],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        694,
        [6, -21, 1, -23, 20, 22, 21, -26],
```

```
  [-24, -4, -46, 40, 44, 42, -52],
  0,
  -1176,
  -1872
],
[
  1331,
  [-6, 5, 19, 22, 3, -17, 14, -40],
  [-8, 44, 44, 6, -34, 28, -80],
  0,
  -4464,
  1488
],
[
  2017,
  [-12, -22, -18, 13, 11, -11, -11, 50],
  [-80, -24, 26, 22, -22, -22, 100],
  0,
  5496,
  -5328
],
[
  2082,
  [-18, -1, -3, -19, -1, -6, -8, 56],
  [-56, 12, -38, -2, -12, -16, 112],
  0,
  6264,
  -3792
],
[
  4083,
  [18, -19, 11, -4, -11, -15, 0, 20],
  [16, 4, -8, -22, -30, 0, 40],
  0,
  1896,
  768
],
[
  5502,
  [-16, -16, 24, 12, -3, 14, 7, -22],
  [-80, 64, 24, -6, 28, 14, -44],
  0,
  -1824,
  -2640
],
[
  6957,
  [16, -14, 15, -11, -7, -23, 12, 12],
  [20, 14, -22, -14, -46, 24, 24],
  0,
  1416,
  1152
],
[
  6961,
  [-10, 5, 22, 7, -12, -6, -18, 12],
  [-20, 54, 14, -24, -12, -36, 24],
  0,
  144,
```

```
  384
],
[
  7420,
  [-16, 18, 15, 11, -22, 11, 5, -22],
  [-12, 46, 22, -44, 22, 10, -44],
  0,
  -2328,
  768
],
[
  7893,
  [-18, 23, -23, 19, -15, 6, -16, 24],
  [-8, -28, 38, -30, 12, -32, 48],
  0,
  2088,
  -1008
],
[
  8136,
  [-14, -1, -2, 12, -24, -15, 16, 28],
  [-44, 10, 24, -48, -30, 32, 56],
  0,
  3816,
  -2496
],
[
  8805,
  [16, 20, -14, 4, -14, -12, -24, 24],
  [88, -44, 8, -28, -24, -48, 48],
  0,
  960,
  3984
],
[
  8876,
  [2, -9, 6, 1, -2, -2, 12, -8],
  [-12, 10, 2, -4, -4, 24, -16],
  0,
  -336,
  -432
],
[
  9600,
  [-14, -13, 8, -22, -4, -7, 16, 36],
  [-68, 30, -44, -8, -14, 32, 72],
  0,
  5160,
  -4080
],
[
  9699,
  [12, 6, 6, -2, -10, 20, -8, -24],
  [48, 0, -4, -20, 40, -16, -48],
  0,
  -3072,
  2928
],
[
```

```
  9838,
  [-14, 1, -7, 23, -19, 4, 4, 8],
  [-40, 0, 46, -38, 8, 8, 16],
  0,
  1248,
  -2064
],
[
  10161,
  [-4, 4, -11, -12, -21, 4, 0, 40],
  [-4, -18, -24, -42, 8, 0, 80],
  0,
  4776,
  -1440
],
[
  10194,
  [-8, 24, 6, -23, 19, -17, 13, -14],
  [24, 20, -46, 38, -34, 26, -28],
  0,
  -1320,
  1632
],
[
  10560,
  [-14, 23, 2, 13, 4, -22, 14, -20],
  [4, 18, 26, 8, -44, 28, -40],
  0,
  -2160,
  1152
],
[
  11018,
  [-24, -6, -23, 23, 21, -13, 14, 8],
  [-84, -22, 46, 42, -26, 28, 16],
  0,
  1896,
  -5040
],
[
  12121,
  [-4, 0, 23, 13, 10, 5, -5, -42],
  [-12, 50, 26, 20, 10, -10, -84],
  0,
  -5184,
  1344
],
[
  12150,
  [16, 18, -9, -15, -13, -17, -24, 44],
  [84, -34, -30, -26, -34, -48, 88],
  0,
  3240,
  3456
],
[
  12565,
  [-20, 12, 15, -9, -24, -7, -13, 46],
  [-36, 50, -18, -48, -14, -26, 92],
```

```
  0,
  4440,
  -1392
],
[
  12620,
  [-24, 22, -5, 10, 23, 12, 6, -44],
  [-28, 14, 20, 46, 24, 12, -88],
  0,
  -4416,
  -624
],
[
  12625,
  [18, -19, 23, -5, 5, 2, -16, -8],
  [16, 28, -10, 10, 4, -32, -16],
  0,
  -1872,
  1776
],
[
  13056,
  [-14, -17, -4, -9, 14, 8, 10, 12],
  [-76, 6, -18, 28, 16, 20, 24],
  0,
  2520,
  -4608
],
[
  13068,
  [-22, -1, -24, -12, 8, -7, 2, 56],
  [-68, -26, -24, 16, -14, 4, 112],
  0,
  6984,
  -5376
],
[
  13883,
  [24, -12, 2, -7, -18, -1, -8, 20],
  [48, -20, -14, -36, -2, -16, 40],
  0,
  1680,
  1920
],
[
  14402,
  [0, -22, -1, 7, 15, 7, -14, 8],
  [-44, -2, 14, 30, 14, -28, 16],
  0,
  600,
  -2400
],
[
  15302,
  [12, 14, -16, 2, 16, 20, 0, -48],
  [64, -44, 4, 32, 40, 0, -96],
  0,
  -5256,
  3072
```

```
      ],
      [
        15485,
        [16, -8, 20, 2, -12, -18, -24, 24],
        [32, 24, 4, -24, -36, -48, 48],
        0,
        864,
        2496
      ],
      [
        15569,
        [-10, 1, -22, -18, 22, -11, 22, 16],
        [-28, -34, -36, 44, -22, 44, 32],
        0,
        3216,
        -3072
      ],
      [
        16143,
        [22, 11, 13, -5, -15, -22, 4, -8],
        [88, 4, -10, -30, -44, 8, -16],
        0,
        -1656,
        5184
      ],
      [
        16213,
        [16, -10, 23, 8, 11, 14, -2, -60],
        [28, 30, 16, 22, 28, -4, -120],
        0,
        -7032,
        3168
      ],
      [
        16459,
        [20, 2, 8, -15, -17, -15, 19, -2],
        [64, -4, -30, -34, -30, 38, -4],
        0,
        144,
        3168
      ],
      [
        17169,
        [12, 2, -20, 23, 1, 1, 7, -26],
        [40, -52, 46, 2, 2, 14, -52],
        0,
        -2712,
        1632
      ],
      [
        17643,
        [16, 16, -20, -9, -14, -17, 24, 4],
        [80, -56, -18, -28, -34, 48, 8],
        0,
        1176,
        2784
      ],
      [
        17646,
```

```
      [12, -16, -3, -17, -18, 5, -9, 46],
      [4, -18, -34, -36, 10, -18, 92],
      0,
      5016,
      -1056
    ]
  ]
},
{
  "depth": 7,
  "odd_part": 3,
  "valid_source_units": 4,
  "trials": 18000,
  "source_coordinates": [
    [0, 1],
    [1, 1],
    [2, 1],
    [3, 1],
    [4, 1],
    [5, 1],
    [6, 1],
    [7, 1],
    [0, 3],
    [1, 3],
    [2, 3],
    [3, 3],
    [4, 3],
    [5, 3],
    [6, 3],
    [7, 3]
  ],
  "target_coordinates": [
    [0, 1],
    [1, 1],
    [2, 1],
    [3, 1],
    [4, 1],
    [5, 1],
    [6, 1],
    [0, 3],
    [1, 3],
    [2, 3],
    [3, 3],
    [4, 3],
    [5, 3],
    [6, 3]
  ],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [
      5283,
      [-7, 20, 0, 18, 11, 17, 13, -8, 3, 5, 20, -18, 24, 18, -22, -94],
      [19, 7, 36, 22, 34, 26, -16, 19, 37, -36, 48, 36, -44, -188],
      0,
      -36888,
      9168
```

```
      ],
      [
        10702,
        [-16, 19, 19, 14, 2, -13, 1, -2, -12, 19, 8, 12, 12, 9, 16, -88],
        [-10, 54, 28, 4, -26, 2, -4, 2, 28, 24, 24, 18, 32, -176],
        0,
        -29184,
        5952
      ],
      [
        12091,
        [18, -21, 10, 22, 13, -3, 17, 12, -10, 3, -1, -17, -20, 16, 9, -48],
        [12, 2, 44, 26, -6, 34, 24, -24, 8, -34, -40, 32, 18, -96],
        0,
        -13632,
        3360
      ],
      [
        12118,
        [-1, 6, 11, 21, -3, 22, 23, 5, -9, -20, 19, 11, -3, 16, 17, -115],
        [9, 23, 42, -6, 44, 46, 10, -67, 47, 22, -6, 32, 34, -230],
        0,
        -36168,
        4128
      ]
    ]
  },
  {
    "depth": 7,
    "odd_part": 5,
    "valid_source_units": 23,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [7, 1],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5],
      [6, 5],
      [7, 5]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [0, 5],
```

```
    [1, 5],
    [2, 5],
    [3, 5],
    [4, 5],
    [5, 5],
    [6, 5]
  ],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [
      895,
      [-13, 19, 10, -15, 9, -1, -10, -7, 21, -11, 11, 7, 17, -1, 3, -39],
      [-1, 33, -30, 18, -2, -20, -14, 41, 1, 14, 34, -2, 6, -78],
      0,
      -23784,
      6048
    ],
    [
      1203,
      [16, -14, 17, 18, 23, -1, 4, -1, 22, 23, -1, 4, 23, -10, -4, -119],
      [20, 18, 36, 46, -2, 8, -2, 112, -24, 8, 46, -20, -8, -238],
      0,
      -76056,
      20592
    ],
    [
      3640,
      [20, -9, -8, -10, 16, -2, -19, 8, -22, 16, 13, -2, -6, 5, 10, -10],
      [42, -36, -20, 32, -4, -38, 16, -34, 48, -4, -12, 10, 20, -20],
      0,
      -2760,
      6240
    ],
    [
      3683,
      [9, 3, -19, 2, -9, 17, 3, -4, 11, 23, -8, -21, 0, -8, 9, -8],
      [33, -47, 4, -18, 34, 6, -8, 79, -27, -42, 0, -16, 18, -16],
      0,
      -4200,
      6816
    ],
    [
      4911,
      [18, 13, -11, 15, -15, 10, -23, -21, -2, 11, 3, 4, 17, -1, -11, -7],
      [80, -40, 30, -30, 20, -46, -42, 16, 8, 8, 34, -2, -22, -14],
      0,
      -10440,
      21744
    ],
    [
      5994,
      [-4, -5, 0, 23, 14, 11, -17, -22, 20, 3, -13, -17, 22, 19, 18, -52],
      [-22, 4, 46, 28, 22, -34, -44, 66, -46, -34, 44, 38, 36, -104],
      0,
      -26688,
      960
```

```
],
[
  7734,
  [21, 14, 7, 18, 0, 24, -21, -19, -11, -5, 3, 9, 13, 10, -24, -39],
  [91, -7, 36, 0, 48, -42, -38, -43, 17, 18, 26, 20, -48, -78],
  0,
  -32472,
  29424
],
[
  7964,
  [-2, 0, -2, -8, 3, -16, -24, -11, 18, 16, -22, -2, 17, -6, 16, 23],
  [-6, -2, -16, 6, -32, -48, -22, 86, -62, -4, 34, -12, 32, 46],
  0,
  16488,
  -720
],
[
  9582,
  [-10, -19, 5, 20, 19, 1, 16, -2, -14, 21, -6, -15, -10, 16, 12, -34],
  [-68, 20, 40, 38, 2, 32, -4, 0, 2, -30, -20, 32, 24, -68],
  0,
  -15504,
  -13968
],
[
  9701,
  [11, -10, -21, 16, -4, 9, 2, -17, 21, 8, 13, -19, -5, -2, 15, -17],
  [13, -53, 32, -8, 18, 4, -34, 79, 5, -38, -10, -4, 30, -34],
  0,
  -8904,
  2688
],
[
  9909,
  [19, -5, 18, -11, 18, 21, 14, -18, -17, -2, 0, -6, -15, -1, 21, -36],
  [47, 17, -22, 36, 42, 28, -36, -55, 17, -12, -30, -2, 42, -72],
  0,
  -18480,
  14976
],
[
  10966,
  [5, -20, -11, -2, -2, -6, 2, 22, -3, -19, -11, -3, -8, 6, 4, 46],
  [-25, -27, -4, -4, -12, 4, 44, -47, -19, -6, -16, 12, 8, 92],
  0,
  33120,
  -16464
],
[
  10993,
  [10, 2, -5, 5, -20, -6, 19, 1, -20, 3, 7, 6, 0, 16, -1, -17],
  [34, -20, 10, -40, -12, 38, 2, -54, 34, 12, 0, 32, -2, -34],
  0,
  -7464,
  4944
],
[
  11169,
```

```
  [4, 19, 13, 5, 17, 21, 21, -12, 18, 24, 22, 14, -2, -21, 1, -144],
  [50, 22, 10, 34, 42, 42, -24, 102, 26, 28, -4, -42, 2, -288],
  0,
  -93144,
  30144
],
[
  11366,
  [2, 4, 8, -19, 5, -1, -6, 5, -24, -11, -3, 5, 14, -6, -12, 39],
  [14, 14, -38, 10, -2, -12, 10, -94, 18, 10, 28, -12, -24, 78],
  0,
  21384,
  -1104
],
[
  11583,
  [-15, -10, 12, 12, 14, 5, 23, -13, 15, 16, -12, 9, 16, -10, 5, -67],
  [-65, 39, 24, 28, 10, 46, -26, 77, -39, 18, 32, -20, 10, -134],
  0,
  -40944,
  -6960
],
[
  11902,
  [-8, 13, 9, 3, -9, 5, 21, 24, 8, -15, -4, -11, -21, 5, 2, -22],
  [2, 26, 6, -18, 10, 42, 48, -6, -16, -22, -42, 10, 4, -44],
  0,
  -10440,
  3840
],
[
  14921,
  [-16, 7, 17, -10, 9, -13, 20, 0, 20, -5, -3, -7, 4, 10, -11, -22],
  [-34, 50, -20, 18, -26, 40, 0, 50, -26, -14, 8, 20, -22, -44],
  0,
  -14976,
  -1680
],
[
  15224,
  [13, 9, 2, -10, 7, 18, 3, -10, -11, 14, -19, 0, 17, -19, 6, -20],
  [57, -9, -20, 14, 36, 6, -20, -5, -27, 0, 34, -38, 12, -40],
  0,
  -13296,
  15360
],
[
  15283,
  [-10, -24, 13, 16, 18, 11, 7, 5, -20, 9, -12, -2, -24, -6, -6, 25],
  [-78, 36, 32, 36, 22, 14, 10, -42, -4, -4, -48, -12, -12, 50],
  0,
  12720,
  -17856
],
[
  15885,
  [2, 20, 1, 6, 9, -17, -13, 12, 8, -13, 9, -21, -2, 13, -12, -2],
  [46, 0, 12, 18, -34, -26, 24, -2, 10, -42, -4, 26, -24, -4],
  0,
```

```
        -3552,
        15120
      ],
      [
        16711,
        [-12, -14, -3, -7, -10, 4, 2, -24, -16, 2, 4, -17, 1, -20, 22, 88],
        [-64, 6, -14, -20, 8, 4, -48, -44, 24, -34, 2, -40, 44, 176],
        0,
        56496,
        -23952
      ],
      [
        16885,
        [9, 1, 9, 8, -6, -11, -20, -24, 11, -23, -14, -13, -4, 8, -9, 78],
        [29, 9, 16, -12, -22, -40, -48, -13, -39, -26, -8, 16, -18, 156],
        0,
        42336,
        8064
      ]
    ]
  },
  {
    "depth": 7,
    "odd_part": 9,
    "valid_source_units": 9,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [7, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [6, 3],
      [7, 3],
      [0, 9],
      [1, 9],
      [2, 9],
      [3, 9],
      [4, 9],
      [5, 9],
      [6, 9],
      [7, 9]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
```

```
    [6, 1],
    [0, 3],
    [1, 3],
    [2, 3],
    [3, 3],
    [4, 3],
    [5, 3],
    [6, 3],
    [0, 9],
    [1, 9],
    [2, 9],
    [3, 9],
    [4, 9],
    [5, 9],
    [6, 9]
  ],
  "columns": [
    "trial", "source_exponents", "target_exponents", "target_weight_sum",
    "target_infinity_sum", "target_reciprocal_sum"
  ],
  "accepted_cases": [
    [
      2724,
      [
        12, -12, 10, -8, -11, 18, 22, 11, -22, -10, 8, -9, 5, -5, -13, 6, 8, 3, -1,
        -23, 7, -14, 15, 3
      ],
      [
        12, 8, -16, -22, 36, 44, 22, -86, 38, -18, 10, -10, -26, 12, 30, -10, -46,
        14, -28, 30, 6
      ],
      0,
      10008,
      -5184
    ],
    [
      8446,
      [
        -22, -1, -2, -4, -24, 17, 21, -15, 12, 15, -13, 3, 1, 8, -23, -3, 16, 23,
        -16, 8, -12, 5, -6, 12
      ],
      [
        -68, 18, -8, -48, 34, 42, -30, 66, -38, 6, 2, 16, -46, -6, 94, -48, 16, -24,
        10, -12, 24
      ],
      0,
      5448,
      -23040
    ],
    [
      9968,
      [
        23, -22, -22, 5, -21, 12, -17, -8, -20, -13, 2, 17, 15, -16, -14, -9, 21,
        -23, -20, -13, 0, 23, 1, 99
      ],
      [
        25, -67, 10, -42, 24, -34, -16, -86, 24, 34, 30, -32, -28, -18, 17, -61,
        -26, 0, 46, 2, 198
      ],
```

```
    0,
    110568,
    -19728
  ],
  [
    10787,
    [
      0, -9, -7, 3, -17, -3, -10, -4, 9, 18, 13, -1, -12, -10, 2, -3, -7, -13, 15,
      6, 1, -5, 11, 23
    ],
    [
      -18, -14, 6, -34, -6, -20, -8, 63, 17, -2, -24, -20, 4, -6, -47, 37, 12, 2,
      -10, 22, 46
    ],
    0,
    28632,
    -5376
  ],
  [
    11393,
    [
      -6, 0, -11, 5, -12, -6, -14, -18, -4, 3, -23, -23, -14, -24, 8, 13, -16,
      -20, 2, -5, 5, 8, 15, 137
    ],
    [
      -18, -16, 10, -24, -12, -28, -36, -6, -42, -46, -28, -48, 16, 26, -88, 20,
      -10, 10, 16, 30, 274
    ],
    0,
    167832,
    -29664
  ],
  [
    14903,
    [
      -19, -13, -16, -9, -16, -1, 10, -9, 12, -14, -21, -4, -10, -8, 13, 20, -3,
      16, 9, 20, -1, -19, -12, 75
    ],
    [
      -83, -13, -18, -32, -2, 20, -18, 8, -54, -8, -20, -16, 26, 40, 23, 21, 40,
      -2, -38, -24, 150
    ],
    0,
    83448,
    -58032
  ],
  [
    15273,
    [
      -22, -19, -16, -7, -23, 22, 16, 10, -11, 9, -9, -11, 10, 8, -11, 7, -17,
      -16, -3, 10, -1, 20, 13, 41
    ],
    [
      -104, -10, -14, -46, 44, 32, 20, -15, -7, -22, 20, 16, -22, 14, -83, 11, 20,
      -2, 40, 26, 82
    ],
    0,
    64440,
    -74160
```

```
      ],
      [
        16306,
        [
          12, 23, 22, 15, 19, -9, 14, 14, -22, 20, -13, -7, -23, -8, 20, 19, 6, -19,
          -7, 16, -23, -9, -15, -45
        ],
        [
          82, 32, 30, 38, -18, 28, 28, -26, -4, -14, -46, -16, 40, 38, -20, -20, 32,
          -46, -18, -30, -90
        ],
        0,
        -53808,
        54624
      ],
      [
        16565,
        [
          -12, -10, 0, 9, -11, 7, 24, -11, -9, -18, -13, -3, -18, -7, 11, 1, 1, 21, 0,
          -10, -9, 7, 8, 42
        ],
        [
          -56, 12, 18, -22, 14, 48, -22, -63, -17, -6, -36, -14, 22, 2, 45, -1, -20,
          -18, 14, 16, 84
        ],
        0,
        54192,
        -38976
      ]
    ]
  },
  {
    "depth": 7,
    "odd_part": 15,
    "valid_source_units": 2,
    "trials": 18000,
    "source_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [7, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [6, 3],
      [7, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5],
```

```
      [6, 5],
      [7, 5],
      [0, 15],
      [1, 15],
      [2, 15],
      [3, 15],
      [4, 15],
      [5, 15],
      [6, 15],
      [7, 15]
    ],
    "target_coordinates": [
      [0, 1],
      [1, 1],
      [2, 1],
      [3, 1],
      [4, 1],
      [5, 1],
      [6, 1],
      [0, 3],
      [1, 3],
      [2, 3],
      [3, 3],
      [4, 3],
      [5, 3],
      [6, 3],
      [0, 5],
      [1, 5],
      [2, 5],
      [3, 5],
      [4, 5],
      [5, 5],
      [6, 5],
      [0, 15],
      [1, 15],
      [2, 15],
      [3, 15],
      [4, 15],
      [5, 15],
      [6, 15]
    ],
    "columns": [
      "trial", "source_exponents", "target_exponents", "target_weight_sum",
      "target_infinity_sum", "target_reciprocal_sum"
    ],
    "accepted_cases": [
      [
        7322,
        [
          -7, -21, 11, 6, 9, 9, 1, -13, 16, 19, -4, 14, -9, 17, -17, 21, -11, -9, -23,
          18, -7, -11, 13, -17, -12, 12, -23, 8, -19, -18, 0, 47
        ],
        [
          -63, 29, 12, 18, 18, 2, -26, 86, -24, 28, -18, 34, -34, 42, -51, -35, 36,
          -14, -22, 26, -34, -12, -34, 16, -38, -36, 0, 94
        ],
        0,
        73200,
        -28944
```

```
        ],
        [
          7388,
          [
            -13, 20, -1, -7, 16, -24, -21, 20, 10, 5, -13, 0, 10, 13, -16, 17, 9, -18,
            -22, 2, 16, 6, -24, 5, -6, 9, -1, -3, 9, 23, -13, -8
          ],
          [
            1, 11, -14, 32, -48, -42, 40, 40, -36, 0, 20, 26, -32, 34, -9, -53, 4, 32,
            12, -48, 10, 0, 4, -6, 18, 46, -26, -16
          ],
          0,
          -11856,
          5712
        ]
      ]
    }
  ],
  "total_valid_source_units": 586,
  "conclusion": "All exact assertions passed."
}
```

A.3. **Odd-prime group rings and marked units.** Every group-ring coefficient for $p = 5, 7, 11, 13$, the character-depth calculation for every prime from 5 through 101, seven complete cusp-order matrices with nonzero determinant witnesses, twenty complete norm matrices with integral preimages, and forty-eight exact atomic resultants. A determinant is recorded by its sign and consecutive decimal digit blocks; concatenating all blocks gives its full absolute value. The output includes a vector with zero first moment and shifted exponents nonzero modulo the coefficient prime. The identities for arbitrary p are (9.10), (9.35), (9.47), (9.41), and (9.42). For the last polynomial, $2(X-1)-(X^2-1) = -(X-1)^2$ has derivative zero at 1 and degree $2 < p-1$ for $p \geq 5$; it is therefore nonzero in $\mathbf{F}_p[X]/((X-1)^{p-1})$. Here $(X-1)\Phi_p(X) = X^p - 1 = (X-1)^p$ in $\mathbf{F}_p[X]$, so $\Phi_p(X) = (X-1)^{p-1}$ in that ring.

Program.

```
"""Exact calculations for the odd-prime seed and marked unit formulas.

The program constructs the group elements, cusp rows, integer matrices,
determinants, integral preimages, and atomic resultants used in the checks.
"""
from pathlib import Path
from collections import Counter
from fractions import Fraction
from math import gcd
import hashlib
import json
import sympy as sp
from sympy.matrices.normalforms import smith_normal_form
from sympy.polys.domains import ZZ

OUT = Path(__file__).with_suffix('.json')
ANS = {"purpose": __doc__, "arithmetic": "exact integers, fractions, finite fields", "tests":
  {}}

def canon(g, p):
    return min(tuple(x % p for x in g), tuple(-x % p for x in g))

def mul(g, h, p):
    a,b,c,d = g
    e,f,k,l = h
    return canon((a*e+b*k,a*f+b*l,c*e+d*k,c*f+d*l),p)

def line(v, p):
    x,y = (v[0] % p, v[1] % p)
    if y == 0:
        return p
    return x * pow(y,-1,p) % p

def sl_projective(p):
    return sorted({canon((a,b,c,d),p)
                   for a in range(p) for b in range(p)
                   for c in range(p) for d in range(p)
                   if (a*d-b*c) % p == 1})

st_results = []
for p in [5,7,11,13]:
    gs = sl_projective(p)
    points = [(x,1) for x in range(p)] + [(1,0)]
    hs = {}
    for g in gs:
```

```
        a,b,c,d = g
        hs[g] = sum(line((a*x+b*y,c*x+d*y),p) == j
                    for j,(x,y) in enumerate(points)) - 1
    conv = dict.fromkeys(gs,0)
    for g in gs:
        if not hs[g]:
            continue
        for h in gs:
            if hs[h]:
                conv[mul(g,h,p)] += hs[g] * hs[h]
    u = len(gs)//p
    bad = [(g,conv[g]-u*hs[g]) for g in gs if conv[g] != u*hs[g]]
    assert not bad and sum(hs.values()) == 0 and gcd(u,p)==1
    st_results.append({"p":p,"group_order":len(gs),"u":u,
        "H_coefficient_distribution":dict(sorted(Counter(hs.values()).items())),
        "augmentation":sum(hs.values()),"H_squared_minus_uH_nonzero_entries":bad,
        "number_of_residual_coefficients_checked":len(gs),
        "coefficient_table_columns":["matrix_a_b_c_d","H","H_squared","H_squared_minus_uH"],
        "complete_coefficient_table":[[list(g),hs[g],conv[g],conv[g]-u*hs[g]] for g in gs]})
ANS["tests"]["steinberg_convolution"] = st_results

def rank_mod(mat, p):
    a = [[x%p for x in row] for row in mat]
    nr,nc = len(a),len(a[0])
    r=0
    for c in range(nc):
        pivot=next((i for i in range(r,nr) if a[i][c]),None)
        if pivot is None:
            continue
        a[r],a[pivot] = a[pivot],a[r]
        inv=pow(a[r][c],-1,p)
        a[r]=[(x*inv)%p for x in a[r]]
        for i in range(nr):
            if i!=r and a[i][c]:
                t=a[i][c]
                a[i]=[(x-t*y)%p for x,y in zip(a[i],a[r])]
        r+=1
        if r==nr:
            break
    return r

char_results=[]
for p in list(sp.primerange(5,102)):
    tail=[(n,d) for n in range(1,p+2) for d in range(p-1)
          if n%(p-1)==0 and (n+d-(p-1))%(p+1)==0]
    igusa=[(n,a) for n in range(1,p+2) for a in range(p-1)
           if n%(p+1)==0 and (n-a)%(p-1)==0]
    assert tail==[(p-1,0)] and igusa==[(p+1,2)]
    scalar_bad=[]
    old_nonscalar=[]
    for a in range(1,p):
        c=a*a%p
        diag=(pow(c,-1,p)*a%p,pow(a,-1,p))
        assert diag[0]==diag[1]
        old=(pow(a,3,p),pow(a,-1,p))
        if old[0]!=old[1]:
            old_nonscalar.append(a)
    # Homogeneous functions f(lambda v)=lambda^2 f(v), the induced character.
    points=[(x,1) for x in range(p)]+[(1,0)]
```

```
    constraints=[]
    for a,b,c,d in [(1,1,0,1),(1,0,1,1)]:
        for j,(x,y) in enumerate(points):
            X,Y=(a*x+b*y)%p,(c*x+d*y)%p
            target=line((X,Y),p)
            scale=Y if Y else X
            row=[0]*(p+1)
            row[target]=scale*scale%p
            row[j]=(row[j]-1)%p
            constraints.append(row)
    rank=rank_mod(constraints,p)
    assert rank==p+1
    char_results.append({"p":p,"tail_solutions":tail,"igusa_solutions":igusa,
        "induced_invariant_dimension":p+1-rank,
        "correct_scalar_bad_residues":scalar_bad,
        "old_scalar_nonscalar_residues":old_nonscalar})
ANS["tests"]["characters"] = char_results

def divs(n):
    return [int(d) for d in sp.divisors(n)]

def cusp_challenge(p,n,M):
    N=p**n*M
    rows=[]
    for i in range(n+1):
        for c0 in divs(M):
            for alpha in (range(1,p) if 0<i<n else [None]):
                c=p**i*c0
                A=1 if alpha is None else alpha*c0%p
                while gcd(A,c)>1:
                    A+=p
                rows.append((i,c0,alpha,A,c))
    columns=[("ordinary",d,None) for d in divs(N) if d!=1]
    columns += [("shift",p**j*d,k) for j in range(n-1)
                for d in divs(M) for k in range(1,p-1)]
    mat=[]
    for i,c0,alpha,A,c in rows:
        width=N//gcd(N,c*c)
        def ordinary(d):
            return Fraction(width*gcd(d,c)**2,d)
        def shifted(d,k):
            return Fraction(width*gcd(p*d*A+k*c,p*c)**2,p*p*d)
        row=[]
        for typ,d,k in columns:
            val=(ordinary(d)-ordinary(1) if typ=="ordinary"
                 else shifted(d,k)-shifted(d,p-1))
            assert val.denominator==1
            row.append(int(val))
        mat.append(row)
    assert len(rows)==(2+(n-1)*(p-1))*len(divs(M))
    assert len(columns)==len(rows)-1
    # Delete the last cusp row: this square minor is an exact rank witness.
    determinant=int(sp.polys.matrices.DomainMatrix.from_Matrix(sp.Matrix(mat[:-1])).det())
    assert determinant!=0
    digits=str(abs(determinant))
    blocks=[digits[i:i+60] for i in range(0,len(digits),60)]
    sign=1 if determinant>0 else -1
    assert sign*int(''.join(blocks))==determinant
    return {"p":p,"n":n,"M":M,"rows":rows,"columns":columns,
```

```
        "matrix":mat,"deleted_row":len(rows)-1,
        "square_minor_determinant":{"sign":sign,
            "decimal_blocks_to_concatenate":blocks},"rank":len(columns)}

ANS["tests"]["cusp_order_exact_rank"]=[cusp_challenge(*case) for case in
    [(5,2,1),(5,2,4),(5,3,6),(7,2,4),(7,3,4),(11,3,1),(13,2,2)]]

norm_stencils=[]
for p in [5,7,11,13]:
    for depth in range(2,7):
        # target b_0,...,b_depth; source a_0,...,a_(depth+1)
        a=sp.zeros(depth+1,depth+2)
        a[0,0]=p+1;a[0,1]=p;a[1,0]=-1;a[1,2]=p
        for i in range(2,depth+1):
            a[i,i+1]=p
        snf=smith_normal_form(a,domain=ZZ)
        diagonals=[abs(int(snf[i,i])) for i in range(depth+1)]
        assert diagonals==[1]+[p]*depth
        assert list(sp.ones(1,depth+1)*a)==[p]*(depth+2)
        target_weights=sp.Matrix([[p**i for i in range(depth+1)]])
        assert list(target_weights*a)==[p**i for i in range(depth+2)]
        # A basis of the complete congruence kernel b0+b1=0; bi=0 (i>=2).
        basis=sp.zeros(depth+1,depth+1)
        basis[0,0]=p;basis[0,1]=-1;basis[1,1]=1
        for i in range(2,depth+1):basis[i,i]=p
        preimages=[]
        for j in range(depth+1):
            b=list(basis[:,j]);v=[0]*(depth+2)
            v[0]=-b[1];v[1]=(b[0]+(p+1)*b[1])//p
            for i in range(2,depth+1):v[i+1]=b[i]//p
            assert a*sp.Matrix(v)==sp.Matrix(b)
            preimages.append(v)
        norm_stencils.append({"p":p,"target_depth":depth,"matrix":a.tolist(),
            "smith_diagonal":diagonals,"congruence_kernel_basis":basis.tolist(),
            "integral_preimage_columns":preimages})
ANS["tests"]["ordinary_norm_stencil"] = norm_stencils

t,q,z=sp.symbols('t q z')
atomic=[]
for p in [5,7]:
    phi=sp.Poly(sum(z**i for i in range(p)),z,domain=sp.ZZ[q])
    for k in range(p):
        for r in [1,2,p,2*p]:
            result=sp.resultant(t**p-q,1-z**k*t**r,t)
            expected=1-q**r if r%p else (1-z**k*q**(r//p))**p
            residue=sp.rem(sp.Poly(sp.expand(result-expected),z,domain=sp.ZZ[q]),phi)
            assert residue.is_zero
            atomic.append({"p":p,"k":k,"r":r,"residual_polynomial":"0"})
ANS["tests"]["atomic_euler_norm_resultants"] = atomic
X=sp.symbols('X')
moment_polynomial=sp.expand(2*(X-1)-(X**2-1))
assert sp.expand(moment_polynomial+(X-1)**2)==0
assert sp.diff(moment_polynomial,X).subs(X,1)==0
assert sp.Poly(moment_polynomial,X).degree()==2
ANS["tests"]["first_moment_counterexample"] = {
    "all_p_ge_5":True,"shift_exponents":[2,-1],
    "first_moment":0,"full_polynomial":"2(X-1)-(X^2-1)=-(X-1)^2",
    "nonzero_mod_Phi_p_mod_p":True,
```

```
    "interpretation":"The first pi moment vanishes while some shifted exponents are nonzero
  modulo p."}

ANS["all_assertions_passed"]=True
ANS["script_sha256"]=hashlib.sha256(Path(__file__).read_bytes()).hexdigest()
OUT.write_text(json.dumps(ANS,ensure_ascii=False,indent=2,default=int)+'\n')
print(json.dumps({"output":str(OUT),"steinberg_cases":len(st_results),
    "character_primes":len(char_results),"cusp_cases":len(ANS['tests']['cusp_order_exact_rank'
  ]),
    "stencil_cases":len(norm_stencils),"atomic_norm_cases":len(atomic),
    "all_assertions_passed":True},ensure_ascii=False))
```

Output.


```
{
  "purpose": "Exact calculations for the odd-prime seed and marked unit formulas.\n\nThe
  program constructs the group elements, cusp rows, integer matrices,\ndeterminants, integral
  preimages, and atomic resultants used in the checks.\n",
  "arithmetic": "exact integers, fractions, finite fields",
  "tests": {
    "steinberg_convolution": [
      {
        "p": 5,
        "group_order": 60,
        "u": 12,
        "H_coefficient_distribution": {"-1": 20, "0": 24, "1": 15, "5": 1},
        "augmentation": 0,
        "H_squared_minus_uH_nonzero_entries": [],
        "number_of_residual_coefficients_checked": 60,
        "coefficient_table_columns": ["matrix_a_b_c_d", "H", "H_squared", "H_squared_minus_uH
  "],
        "complete_coefficient_table": [
          [[0, 1, 4, 0], 1, 12, 0],
          [[0, 1, 4, 1], -1, -12, 0],
          [[0, 1, 4, 2], 0, 0, 0],
          [[0, 1, 4, 3], 0, 0, 0],
          [[0, 1, 4, 4], -1, -12, 0],
          [[0, 2, 2, 0], 1, 12, 0],
          [[0, 2, 2, 1], -1, -12, 0],
          [[0, 2, 2, 2], 0, 0, 0],
          [[0, 2, 2, 3], 0, 0, 0],
          [[0, 2, 2, 4], -1, -12, 0],
          [[1, 0, 0, 1], 5, 60, 0],
          [[1, 0, 1, 1], 0, 0, 0],
          [[1, 0, 2, 1], 0, 0, 0],
          [[1, 0, 3, 1], 0, 0, 0],
          [[1, 0, 4, 1], 0, 0, 0],
          [[1, 1, 0, 1], 0, 0, 0],
          [[1, 1, 1, 2], 0, 0, 0],
          [[1, 1, 2, 3], -1, -12, 0],
          [[1, 1, 3, 4], 1, 12, 0],
          [[1, 1, 4, 0], -1, -12, 0],
          [[1, 2, 0, 1], 0, 0, 0],
          [[1, 2, 1, 3], -1, -12, 0],
          [[1, 2, 2, 0], -1, -12, 0],
          [[1, 2, 3, 2], 0, 0, 0],
          [[1, 2, 4, 4], 1, 12, 0],
          [[1, 3, 0, 1], 0, 0, 0],
          [[1, 3, 1, 4], 1, 12, 0],
```

```
        [[1, 3, 2, 2], 0, 0, 0],
        [[1, 3, 3, 0], -1, -12, 0],
        [[1, 3, 4, 3], -1, -12, 0],
        [[1, 4, 0, 1], 0, 0, 0],
        [[1, 4, 1, 0], -1, -12, 0],
        [[1, 4, 2, 4], 1, 12, 0],
        [[1, 4, 3, 3], -1, -12, 0],
        [[1, 4, 4, 2], 0, 0, 0],
        [[2, 0, 0, 3], 1, 12, 0],
        [[2, 0, 1, 3], 1, 12, 0],
        [[2, 0, 2, 3], 1, 12, 0],
        [[2, 0, 3, 3], 1, 12, 0],
        [[2, 0, 4, 3], 1, 12, 0],
        [[2, 1, 0, 3], 1, 12, 0],
        [[2, 1, 1, 1], 0, 0, 0],
        [[2, 1, 2, 4], -1, -12, 0],
        [[2, 1, 3, 2], -1, -12, 0],
        [[2, 1, 4, 0], 0, 0, 0],
        [[2, 2, 0, 3], 1, 12, 0],
        [[2, 2, 1, 4], -1, -12, 0],
        [[2, 2, 2, 0], 0, 0, 0],
        [[2, 2, 3, 1], 0, 0, 0],
        [[2, 2, 4, 2], -1, -12, 0],
        [[2, 3, 0, 3], 1, 12, 0],
        [[2, 3, 1, 2], -1, -12, 0],
        [[2, 3, 2, 1], 0, 0, 0],
        [[2, 3, 3, 0], 0, 0, 0],
        [[2, 3, 4, 4], -1, -12, 0],
        [[2, 4, 0, 3], 1, 12, 0],
        [[2, 4, 1, 0], 0, 0, 0],
        [[2, 4, 2, 2], -1, -12, 0],
        [[2, 4, 3, 4], -1, -12, 0],
        [[2, 4, 4, 1], 0, 0, 0]
      ]
    },
    {
      "p": 7,
      "group_order": 168,
      "u": 24,
      "H_coefficient_distribution": {"-1": 63, "0": 48, "1": 56, "7": 1},
      "augmentation": 0,
      "H_squared_minus_uH_nonzero_entries": [],
      "number_of_residual_coefficients_checked": 168,
      "coefficient_table_columns": ["matrix_a_b_c_d", "H", "H_squared", "H_squared_minus_uH
"],
      "complete_coefficient_table": [
        [[0, 1, 6, 0], -1, -24, 0],
        [[0, 1, 6, 1], 1, 24, 0],
        [[0, 1, 6, 2], 0, 0, 0],
        [[0, 1, 6, 3], -1, -24, 0],
        [[0, 1, 6, 4], -1, -24, 0],
        [[0, 1, 6, 5], 0, 0, 0],
        [[0, 1, 6, 6], 1, 24, 0],
        [[0, 2, 3, 0], -1, -24, 0],
        [[0, 2, 3, 1], 1, 24, 0],
        [[0, 2, 3, 2], 0, 0, 0],
        [[0, 2, 3, 3], -1, -24, 0],
        [[0, 2, 3, 4], -1, -24, 0],
        [[0, 2, 3, 5], 0, 0, 0],
```

```
[[0, 2, 3, 6], 1, 24, 0],
[[0, 3, 2, 0], -1, -24, 0],
[[0, 3, 2, 1], 1, 24, 0],
[[0, 3, 2, 2], 0, 0, 0],
[[0, 3, 2, 3], -1, -24, 0],
[[0, 3, 2, 4], -1, -24, 0],
[[0, 3, 2, 5], 0, 0, 0],
[[0, 3, 2, 6], 1, 24, 0],
[[1, 0, 0, 1], 7, 168, 0],
[[1, 0, 1, 1], 0, 0, 0],
[[1, 0, 2, 1], 0, 0, 0],
[[1, 0, 3, 1], 0, 0, 0],
[[1, 0, 4, 1], 0, 0, 0],
[[1, 0, 5, 1], 0, 0, 0],
[[1, 0, 6, 1], 0, 0, 0],
[[1, 1, 0, 1], 0, 0, 0],
[[1, 1, 1, 2], -1, -24, 0],
[[1, 1, 2, 3], -1, -24, 0],
[[1, 1, 3, 4], 0, 0, 0],
[[1, 1, 4, 5], 1, 24, 0],
[[1, 1, 5, 6], -1, -24, 0],
[[1, 1, 6, 0], 1, 24, 0],
[[1, 2, 0, 1], 0, 0, 0],
[[1, 2, 1, 3], -1, -24, 0],
[[1, 2, 2, 5], 1, 24, 0],
[[1, 2, 3, 0], 1, 24, 0],
[[1, 2, 4, 2], -1, -24, 0],
[[1, 2, 5, 4], 0, 0, 0],
[[1, 2, 6, 6], -1, -24, 0],
[[1, 3, 0, 1], 0, 0, 0],
[[1, 3, 1, 4], 0, 0, 0],
[[1, 3, 2, 0], 1, 24, 0],
[[1, 3, 3, 3], -1, -24, 0],
[[1, 3, 4, 6], -1, -24, 0],
[[1, 3, 5, 2], -1, -24, 0],
[[1, 3, 6, 5], 1, 24, 0],
[[1, 4, 0, 1], 0, 0, 0],
[[1, 4, 1, 5], 1, 24, 0],
[[1, 4, 2, 2], -1, -24, 0],
[[1, 4, 3, 6], -1, -24, 0],
[[1, 4, 4, 3], -1, -24, 0],
[[1, 4, 5, 0], 1, 24, 0],
[[1, 4, 6, 4], 0, 0, 0],
[[1, 5, 0, 1], 0, 0, 0],
[[1, 5, 1, 6], -1, -24, 0],
[[1, 5, 2, 4], 0, 0, 0],
[[1, 5, 3, 2], -1, -24, 0],
[[1, 5, 4, 0], 1, 24, 0],
[[1, 5, 5, 5], 1, 24, 0],
[[1, 5, 6, 3], -1, -24, 0],
[[1, 6, 0, 1], 0, 0, 0],
[[1, 6, 1, 0], 1, 24, 0],
[[1, 6, 2, 6], -1, -24, 0],
[[1, 6, 3, 5], 1, 24, 0],
[[1, 6, 4, 4], 0, 0, 0],
[[1, 6, 5, 3], -1, -24, 0],
[[1, 6, 6, 2], -1, -24, 0],
[[2, 0, 0, 4], 1, 24, 0],
[[2, 0, 1, 4], 1, 24, 0],
```

```
[[2, 0, 2, 4], 1, 24, 0],
[[2, 0, 3, 4], 1, 24, 0],
[[2, 0, 4, 4], 1, 24, 0],
[[2, 0, 5, 4], 1, 24, 0],
[[2, 0, 6, 4], 1, 24, 0],
[[2, 1, 0, 4], 1, 24, 0],
[[2, 1, 1, 1], -1, -24, 0],
[[2, 1, 2, 5], -1, -24, 0],
[[2, 1, 3, 2], -1, -24, 0],
[[2, 1, 4, 6], 1, 24, 0],
[[2, 1, 5, 3], 0, 0, 0],
[[2, 1, 6, 0], 0, 0, 0],
[[2, 2, 0, 4], 1, 24, 0],
[[2, 2, 1, 5], -1, -24, 0],
[[2, 2, 2, 6], 1, 24, 0],
[[2, 2, 3, 0], 0, 0, 0],
[[2, 2, 4, 1], -1, -24, 0],
[[2, 2, 5, 2], -1, -24, 0],
[[2, 2, 6, 3], 0, 0, 0],
[[2, 3, 0, 4], 1, 24, 0],
[[2, 3, 1, 2], -1, -24, 0],
[[2, 3, 2, 0], 0, 0, 0],
[[2, 3, 3, 5], -1, -24, 0],
[[2, 3, 4, 3], 0, 0, 0],
[[2, 3, 5, 1], -1, -24, 0],
[[2, 3, 6, 6], 1, 24, 0],
[[2, 4, 0, 4], 1, 24, 0],
[[2, 4, 1, 6], 1, 24, 0],
[[2, 4, 2, 1], -1, -24, 0],
[[2, 4, 3, 3], 0, 0, 0],
[[2, 4, 4, 5], -1, -24, 0],
[[2, 4, 5, 0], 0, 0, 0],
[[2, 4, 6, 2], -1, -24, 0],
[[2, 5, 0, 4], 1, 24, 0],
[[2, 5, 1, 3], 0, 0, 0],
[[2, 5, 2, 2], -1, -24, 0],
[[2, 5, 3, 1], -1, -24, 0],
[[2, 5, 4, 0], 0, 0, 0],
[[2, 5, 5, 6], 1, 24, 0],
[[2, 5, 6, 5], -1, -24, 0],
[[2, 6, 0, 4], 1, 24, 0],
[[2, 6, 1, 0], 0, 0, 0],
[[2, 6, 2, 3], 0, 0, 0],
[[2, 6, 3, 6], 1, 24, 0],
[[2, 6, 4, 2], -1, -24, 0],
[[2, 6, 5, 5], -1, -24, 0],
[[2, 6, 6, 1], -1, -24, 0],
[[3, 0, 0, 5], 1, 24, 0],
[[3, 0, 1, 5], 1, 24, 0],
[[3, 0, 2, 5], 1, 24, 0],
[[3, 0, 3, 5], 1, 24, 0],
[[3, 0, 4, 5], 1, 24, 0],
[[3, 0, 5, 5], 1, 24, 0],
[[3, 0, 6, 5], 1, 24, 0],
[[3, 1, 0, 5], 1, 24, 0],
[[3, 1, 1, 3], 1, 24, 0],
[[3, 1, 2, 1], -1, -24, 0],
[[3, 1, 3, 6], 0, 0, 0],
[[3, 1, 4, 4], -1, -24, 0],
```

```
        [[3, 1, 5, 2], 0, 0, 0],
        [[3, 1, 6, 0], -1, -24, 0],
        [[3, 2, 0, 5], 1, 24, 0],
        [[3, 2, 1, 1], -1, -24, 0],
        [[3, 2, 2, 4], -1, -24, 0],
        [[3, 2, 3, 0], -1, -24, 0],
        [[3, 2, 4, 3], 1, 24, 0],
        [[3, 2, 5, 6], 0, 0, 0],
        [[3, 2, 6, 2], 0, 0, 0],
        [[3, 3, 0, 5], 1, 24, 0],
        [[3, 3, 1, 6], 0, 0, 0],
        [[3, 3, 2, 0], -1, -24, 0],
        [[3, 3, 3, 1], -1, -24, 0],
        [[3, 3, 4, 2], 0, 0, 0],
        [[3, 3, 5, 3], 1, 24, 0],
        [[3, 3, 6, 4], -1, -24, 0],
        [[3, 4, 0, 5], 1, 24, 0],
        [[3, 4, 1, 4], -1, -24, 0],
        [[3, 4, 2, 3], 1, 24, 0],
        [[3, 4, 3, 2], 0, 0, 0],
        [[3, 4, 4, 1], -1, -24, 0],
        [[3, 4, 5, 0], -1, -24, 0],
        [[3, 4, 6, 6], 0, 0, 0],
        [[3, 5, 0, 5], 1, 24, 0],
        [[3, 5, 1, 2], 0, 0, 0],
        [[3, 5, 2, 6], 0, 0, 0],
        [[3, 5, 3, 3], 1, 24, 0],
        [[3, 5, 4, 0], -1, -24, 0],
        [[3, 5, 5, 4], -1, -24, 0],
        [[3, 5, 6, 1], -1, -24, 0],
        [[3, 6, 0, 5], 1, 24, 0],
        [[3, 6, 1, 0], -1, -24, 0],
        [[3, 6, 2, 2], 0, 0, 0],
        [[3, 6, 3, 4], -1, -24, 0],
        [[3, 6, 4, 6], 0, 0, 0],
        [[3, 6, 5, 1], -1, -24, 0],
        [[3, 6, 6, 3], 1, 24, 0]
      ]
    },
    {
      "p": 11,
      "group_order": 660,
      "u": 60,
      "H_coefficient_distribution": {"-1": 275, "0": 120, "1": 264, "11": 1},
      "augmentation": 0,
      "H_squared_minus_uH_nonzero_entries": [],
      "number_of_residual_coefficients_checked": 660,
      "coefficient_table_columns": ["matrix_a_b_c_d", "H", "H_squared", "H_squared_minus_uH
"],
      "complete_coefficient_table": [
        [[0, 1, 10, 0], -1, -60, 0],
        [[0, 1, 10, 1], -1, -60, 0],
        [[0, 1, 10, 2], 0, 0, 0],
        [[0, 1, 10, 3], 1, 60, 0],
        [[0, 1, 10, 4], 1, 60, 0],
        [[0, 1, 10, 5], -1, -60, 0],
        [[0, 1, 10, 6], -1, -60, 0],
        [[0, 1, 10, 7], 1, 60, 0],
        [[0, 1, 10, 8], 1, 60, 0],
```

```
[[0, 1, 10, 9], 0, 0, 0],
[[0, 1, 10, 10], -1, -60, 0],
[[0, 2, 5, 0], -1, -60, 0],
[[0, 2, 5, 1], -1, -60, 0],
[[0, 2, 5, 2], 0, 0, 0],
[[0, 2, 5, 3], 1, 60, 0],
[[0, 2, 5, 4], 1, 60, 0],
[[0, 2, 5, 5], -1, -60, 0],
[[0, 2, 5, 6], -1, -60, 0],
[[0, 2, 5, 7], 1, 60, 0],
[[0, 2, 5, 8], 1, 60, 0],
[[0, 2, 5, 9], 0, 0, 0],
[[0, 2, 5, 10], -1, -60, 0],
[[0, 3, 7, 0], -1, -60, 0],
[[0, 3, 7, 1], -1, -60, 0],
[[0, 3, 7, 2], 0, 0, 0],
[[0, 3, 7, 3], 1, 60, 0],
[[0, 3, 7, 4], 1, 60, 0],
[[0, 3, 7, 5], -1, -60, 0],
[[0, 3, 7, 6], -1, -60, 0],
[[0, 3, 7, 7], 1, 60, 0],
[[0, 3, 7, 8], 1, 60, 0],
[[0, 3, 7, 9], 0, 0, 0],
[[0, 3, 7, 10], -1, -60, 0],
[[0, 4, 8, 0], -1, -60, 0],
[[0, 4, 8, 1], -1, -60, 0],
[[0, 4, 8, 2], 0, 0, 0],
[[0, 4, 8, 3], 1, 60, 0],
[[0, 4, 8, 4], 1, 60, 0],
[[0, 4, 8, 5], -1, -60, 0],
[[0, 4, 8, 6], -1, -60, 0],
[[0, 4, 8, 7], 1, 60, 0],
[[0, 4, 8, 8], 1, 60, 0],
[[0, 4, 8, 9], 0, 0, 0],
[[0, 4, 8, 10], -1, -60, 0],
[[0, 5, 2, 0], -1, -60, 0],
[[0, 5, 2, 1], -1, -60, 0],
[[0, 5, 2, 2], 0, 0, 0],
[[0, 5, 2, 3], 1, 60, 0],
[[0, 5, 2, 4], 1, 60, 0],
[[0, 5, 2, 5], -1, -60, 0],
[[0, 5, 2, 6], -1, -60, 0],
[[0, 5, 2, 7], 1, 60, 0],
[[0, 5, 2, 8], 1, 60, 0],
[[0, 5, 2, 9], 0, 0, 0],
[[0, 5, 2, 10], -1, -60, 0],
[[1, 0, 0, 1], 11, 660, 0],
[[1, 0, 1, 1], 0, 0, 0],
[[1, 0, 2, 1], 0, 0, 0],
[[1, 0, 3, 1], 0, 0, 0],
[[1, 0, 4, 1], 0, 0, 0],
[[1, 0, 5, 1], 0, 0, 0],
[[1, 0, 6, 1], 0, 0, 0],
[[1, 0, 7, 1], 0, 0, 0],
[[1, 0, 8, 1], 0, 0, 0],
[[1, 0, 9, 1], 0, 0, 0],
[[1, 0, 10, 1], 0, 0, 0],
[[1, 1, 0, 1], 0, 0, 0],
[[1, 1, 1, 2], 1, 60, 0],
```

```
[[1, 1, 2, 3], 1, 60, 0],
[[1, 1, 3, 4], -1, -60, 0],
[[1, 1, 4, 5], -1, -60, 0],
[[1, 1, 5, 6], 1, 60, 0],
[[1, 1, 6, 7], 1, 60, 0],
[[1, 1, 7, 8], 0, 0, 0],
[[1, 1, 8, 9], -1, -60, 0],
[[1, 1, 9, 10], -1, -60, 0],
[[1, 1, 10, 0], -1, -60, 0],
[[1, 2, 0, 1], 0, 0, 0],
[[1, 2, 1, 3], 1, 60, 0],
[[1, 2, 2, 5], -1, -60, 0],
[[1, 2, 3, 7], 1, 60, 0],
[[1, 2, 4, 9], -1, -60, 0],
[[1, 2, 5, 0], -1, -60, 0],
[[1, 2, 6, 2], 1, 60, 0],
[[1, 2, 7, 4], -1, -60, 0],
[[1, 2, 8, 6], 1, 60, 0],
[[1, 2, 9, 8], 0, 0, 0],
[[1, 2, 10, 10], -1, -60, 0],
[[1, 3, 0, 1], 0, 0, 0],
[[1, 3, 1, 4], -1, -60, 0],
[[1, 3, 2, 7], 1, 60, 0],
[[1, 3, 3, 10], -1, -60, 0],
[[1, 3, 4, 2], 1, 60, 0],
[[1, 3, 5, 5], -1, -60, 0],
[[1, 3, 6, 8], 0, 0, 0],
[[1, 3, 7, 0], -1, -60, 0],
[[1, 3, 8, 3], 1, 60, 0],
[[1, 3, 9, 6], 1, 60, 0],
[[1, 3, 10, 9], -1, -60, 0],
[[1, 4, 0, 1], 0, 0, 0],
[[1, 4, 1, 5], -1, -60, 0],
[[1, 4, 2, 9], -1, -60, 0],
[[1, 4, 3, 2], 1, 60, 0],
[[1, 4, 4, 6], 1, 60, 0],
[[1, 4, 5, 10], -1, -60, 0],
[[1, 4, 6, 3], 1, 60, 0],
[[1, 4, 7, 7], 1, 60, 0],
[[1, 4, 8, 0], -1, -60, 0],
[[1, 4, 9, 4], -1, -60, 0],
[[1, 4, 10, 8], 0, 0, 0],
[[1, 5, 0, 1], 0, 0, 0],
[[1, 5, 1, 6], 1, 60, 0],
[[1, 5, 2, 0], -1, -60, 0],
[[1, 5, 3, 5], -1, -60, 0],
[[1, 5, 4, 10], -1, -60, 0],
[[1, 5, 5, 4], -1, -60, 0],
[[1, 5, 6, 9], -1, -60, 0],
[[1, 5, 7, 3], 1, 60, 0],
[[1, 5, 8, 8], 0, 0, 0],
[[1, 5, 9, 2], 1, 60, 0],
[[1, 5, 10, 7], 1, 60, 0],
[[1, 6, 0, 1], 0, 0, 0],
[[1, 6, 1, 7], 1, 60, 0],
[[1, 6, 2, 2], 1, 60, 0],
[[1, 6, 3, 8], 0, 0, 0],
[[1, 6, 4, 3], 1, 60, 0],
[[1, 6, 5, 9], -1, -60, 0],
```

```
[[1, 6, 6, 4], -1, -60, 0],
[[1, 6, 7, 10], -1, -60, 0],
[[1, 6, 8, 5], -1, -60, 0],
[[1, 6, 9, 0], -1, -60, 0],
[[1, 6, 10, 6], 1, 60, 0],
[[1, 7, 0, 1], 0, 0, 0],
[[1, 7, 1, 8], 0, 0, 0],
[[1, 7, 2, 4], -1, -60, 0],
[[1, 7, 3, 0], -1, -60, 0],
[[1, 7, 4, 7], 1, 60, 0],
[[1, 7, 5, 3], 1, 60, 0],
[[1, 7, 6, 10], -1, -60, 0],
[[1, 7, 7, 6], 1, 60, 0],
[[1, 7, 8, 2], 1, 60, 0],
[[1, 7, 9, 9], -1, -60, 0],
[[1, 7, 10, 5], -1, -60, 0],
[[1, 8, 0, 1], 0, 0, 0],
[[1, 8, 1, 9], -1, -60, 0],
[[1, 8, 2, 6], 1, 60, 0],
[[1, 8, 3, 3], 1, 60, 0],
[[1, 8, 4, 0], -1, -60, 0],
[[1, 8, 5, 8], 0, 0, 0],
[[1, 8, 6, 5], -1, -60, 0],
[[1, 8, 7, 2], 1, 60, 0],
[[1, 8, 8, 10], -1, -60, 0],
[[1, 8, 9, 7], 1, 60, 0],
[[1, 8, 10, 4], -1, -60, 0],
[[1, 9, 0, 1], 0, 0, 0],
[[1, 9, 1, 10], -1, -60, 0],
[[1, 9, 2, 8], 0, 0, 0],
[[1, 9, 3, 6], 1, 60, 0],
[[1, 9, 4, 4], -1, -60, 0],
[[1, 9, 5, 2], 1, 60, 0],
[[1, 9, 6, 0], -1, -60, 0],
[[1, 9, 7, 9], -1, -60, 0],
[[1, 9, 8, 7], 1, 60, 0],
[[1, 9, 9, 5], -1, -60, 0],
[[1, 9, 10, 3], 1, 60, 0],
[[1, 10, 0, 1], 0, 0, 0],
[[1, 10, 1, 0], -1, -60, 0],
[[1, 10, 2, 10], -1, -60, 0],
[[1, 10, 3, 9], -1, -60, 0],
[[1, 10, 4, 8], 0, 0, 0],
[[1, 10, 5, 7], 1, 60, 0],
[[1, 10, 6, 6], 1, 60, 0],
[[1, 10, 7, 5], -1, -60, 0],
[[1, 10, 8, 4], -1, -60, 0],
[[1, 10, 9, 3], 1, 60, 0],
[[1, 10, 10, 2], 1, 60, 0],
[[2, 0, 0, 6], 1, 60, 0],
[[2, 0, 1, 6], 1, 60, 0],
[[2, 0, 2, 6], 1, 60, 0],
[[2, 0, 3, 6], 1, 60, 0],
[[2, 0, 4, 6], 1, 60, 0],
[[2, 0, 5, 6], 1, 60, 0],
[[2, 0, 6, 6], 1, 60, 0],
[[2, 0, 7, 6], 1, 60, 0],
[[2, 0, 8, 6], 1, 60, 0],
[[2, 0, 9, 6], 1, 60, 0],
```

```
[[2, 0, 10, 6], 1, 60, 0],
[[2, 1, 0, 6], 1, 60, 0],
[[2, 1, 1, 1], 1, 60, 0],
[[2, 1, 2, 7], 0, 0, 0],
[[2, 1, 3, 2], 1, 60, 0],
[[2, 1, 4, 8], -1, -60, 0],
[[2, 1, 5, 3], -1, -60, 0],
[[2, 1, 6, 9], -1, -60, 0],
[[2, 1, 7, 4], -1, -60, 0],
[[2, 1, 8, 10], -1, -60, 0],
[[2, 1, 9, 5], 1, 60, 0],
[[2, 1, 10, 0], 0, 0, 0],
[[2, 2, 0, 6], 1, 60, 0],
[[2, 2, 1, 7], 0, 0, 0],
[[2, 2, 2, 8], -1, -60, 0],
[[2, 2, 3, 9], -1, -60, 0],
[[2, 2, 4, 10], -1, -60, 0],
[[2, 2, 5, 0], 0, 0, 0],
[[2, 2, 6, 1], 1, 60, 0],
[[2, 2, 7, 2], 1, 60, 0],
[[2, 2, 8, 3], -1, -60, 0],
[[2, 2, 9, 4], -1, -60, 0],
[[2, 2, 10, 5], 1, 60, 0],
[[2, 3, 0, 6], 1, 60, 0],
[[2, 3, 1, 2], 1, 60, 0],
[[2, 3, 2, 9], -1, -60, 0],
[[2, 3, 3, 5], 1, 60, 0],
[[2, 3, 4, 1], 1, 60, 0],
[[2, 3, 5, 8], -1, -60, 0],
[[2, 3, 6, 4], -1, -60, 0],
[[2, 3, 7, 0], 0, 0, 0],
[[2, 3, 8, 7], 0, 0, 0],
[[2, 3, 9, 3], -1, -60, 0],
[[2, 3, 10, 10], -1, -60, 0],
[[2, 4, 0, 6], 1, 60, 0],
[[2, 4, 1, 8], -1, -60, 0],
[[2, 4, 2, 10], -1, -60, 0],
[[2, 4, 3, 1], 1, 60, 0],
[[2, 4, 4, 3], -1, -60, 0],
[[2, 4, 5, 5], 1, 60, 0],
[[2, 4, 6, 7], 0, 0, 0],
[[2, 4, 7, 9], -1, -60, 0],
[[2, 4, 8, 0], 0, 0, 0],
[[2, 4, 9, 2], 1, 60, 0],
[[2, 4, 10, 4], -1, -60, 0],
[[2, 5, 0, 6], 1, 60, 0],
[[2, 5, 1, 3], -1, -60, 0],
[[2, 5, 2, 0], 0, 0, 0],
[[2, 5, 3, 8], -1, -60, 0],
[[2, 5, 4, 5], 1, 60, 0],
[[2, 5, 5, 2], 1, 60, 0],
[[2, 5, 6, 10], -1, -60, 0],
[[2, 5, 7, 7], 0, 0, 0],
[[2, 5, 8, 4], -1, -60, 0],
[[2, 5, 9, 1], 1, 60, 0],
[[2, 5, 10, 9], -1, -60, 0],
[[2, 6, 0, 6], 1, 60, 0],
[[2, 6, 1, 9], -1, -60, 0],
[[2, 6, 2, 1], 1, 60, 0],
```

```
[[2, 6, 3, 4], -1, -60, 0],
[[2, 6, 4, 7], 0, 0, 0],
[[2, 6, 5, 10], -1, -60, 0],
[[2, 6, 6, 2], 1, 60, 0],
[[2, 6, 7, 5], 1, 60, 0],
[[2, 6, 8, 8], -1, -60, 0],
[[2, 6, 9, 0], 0, 0, 0],
[[2, 6, 10, 3], -1, -60, 0],
[[2, 7, 0, 6], 1, 60, 0],
[[2, 7, 1, 4], -1, -60, 0],
[[2, 7, 2, 2], 1, 60, 0],
[[2, 7, 3, 0], 0, 0, 0],
[[2, 7, 4, 9], -1, -60, 0],
[[2, 7, 5, 7], 0, 0, 0],
[[2, 7, 6, 5], 1, 60, 0],
[[2, 7, 7, 3], -1, -60, 0],
[[2, 7, 8, 1], 1, 60, 0],
[[2, 7, 9, 10], -1, -60, 0],
[[2, 7, 10, 8], -1, -60, 0],
[[2, 8, 0, 6], 1, 60, 0],
[[2, 8, 1, 10], -1, -60, 0],
[[2, 8, 2, 3], -1, -60, 0],
[[2, 8, 3, 7], 0, 0, 0],
[[2, 8, 4, 0], 0, 0, 0],
[[2, 8, 5, 4], -1, -60, 0],
[[2, 8, 6, 8], -1, -60, 0],
[[2, 8, 7, 1], 1, 60, 0],
[[2, 8, 8, 5], 1, 60, 0],
[[2, 8, 9, 9], -1, -60, 0],
[[2, 8, 10, 2], 1, 60, 0],
[[2, 9, 0, 6], 1, 60, 0],
[[2, 9, 1, 5], 1, 60, 0],
[[2, 9, 2, 4], -1, -60, 0],
[[2, 9, 3, 3], -1, -60, 0],
[[2, 9, 4, 2], 1, 60, 0],
[[2, 9, 5, 1], 1, 60, 0],
[[2, 9, 6, 0], 0, 0, 0],
[[2, 9, 7, 10], -1, -60, 0],
[[2, 9, 8, 9], -1, -60, 0],
[[2, 9, 9, 8], -1, -60, 0],
[[2, 9, 10, 7], 0, 0, 0],
[[2, 10, 0, 6], 1, 60, 0],
[[2, 10, 1, 0], 0, 0, 0],
[[2, 10, 2, 5], 1, 60, 0],
[[2, 10, 3, 10], -1, -60, 0],
[[2, 10, 4, 4], -1, -60, 0],
[[2, 10, 5, 9], -1, -60, 0],
[[2, 10, 6, 3], -1, -60, 0],
[[2, 10, 7, 8], -1, -60, 0],
[[2, 10, 8, 2], 1, 60, 0],
[[2, 10, 9, 7], 0, 0, 0],
[[2, 10, 10, 1], 1, 60, 0],
[[3, 0, 0, 4], 1, 60, 0],
[[3, 0, 1, 4], 1, 60, 0],
[[3, 0, 2, 4], 1, 60, 0],
[[3, 0, 3, 4], 1, 60, 0],
[[3, 0, 4, 4], 1, 60, 0],
[[3, 0, 5, 4], 1, 60, 0],
[[3, 0, 6, 4], 1, 60, 0],
```

```
[[3, 0, 7, 4], 1, 60, 0],
[[3, 0, 8, 4], 1, 60, 0],
[[3, 0, 9, 4], 1, 60, 0],
[[3, 0, 10, 4], 1, 60, 0],
[[3, 1, 0, 4], 1, 60, 0],
[[3, 1, 1, 8], -1, -60, 0],
[[3, 1, 2, 1], 1, 60, 0],
[[3, 1, 3, 5], 1, 60, 0],
[[3, 1, 4, 9], -1, -60, 0],
[[3, 1, 5, 2], -1, -60, 0],
[[3, 1, 6, 6], 0, 0, 0],
[[3, 1, 7, 10], 0, 0, 0],
[[3, 1, 8, 3], -1, -60, 0],
[[3, 1, 9, 7], -1, -60, 0],
[[3, 1, 10, 0], 1, 60, 0],
[[3, 2, 0, 4], 1, 60, 0],
[[3, 2, 1, 1], 1, 60, 0],
[[3, 2, 2, 9], -1, -60, 0],
[[3, 2, 3, 6], 0, 0, 0],
[[3, 2, 4, 3], -1, -60, 0],
[[3, 2, 5, 0], 1, 60, 0],
[[3, 2, 6, 8], -1, -60, 0],
[[3, 2, 7, 5], 1, 60, 0],
[[3, 2, 8, 2], -1, -60, 0],
[[3, 2, 9, 10], 0, 0, 0],
[[3, 2, 10, 7], -1, -60, 0],
[[3, 3, 0, 4], 1, 60, 0],
[[3, 3, 1, 5], 1, 60, 0],
[[3, 3, 2, 6], 0, 0, 0],
[[3, 3, 3, 7], -1, -60, 0],
[[3, 3, 4, 8], -1, -60, 0],
[[3, 3, 5, 9], -1, -60, 0],
[[3, 3, 6, 10], 0, 0, 0],
[[3, 3, 7, 0], 1, 60, 0],
[[3, 3, 8, 1], 1, 60, 0],
[[3, 3, 9, 2], -1, -60, 0],
[[3, 3, 10, 3], -1, -60, 0],
[[3, 4, 0, 4], 1, 60, 0],
[[3, 4, 1, 9], -1, -60, 0],
[[3, 4, 2, 3], -1, -60, 0],
[[3, 4, 3, 8], -1, -60, 0],
[[3, 4, 4, 2], -1, -60, 0],
[[3, 4, 5, 7], -1, -60, 0],
[[3, 4, 6, 1], 1, 60, 0],
[[3, 4, 7, 6], 0, 0, 0],
[[3, 4, 8, 0], 1, 60, 0],
[[3, 4, 9, 5], 1, 60, 0],
[[3, 4, 10, 10], 0, 0, 0],
[[3, 5, 0, 4], 1, 60, 0],
[[3, 5, 1, 2], -1, -60, 0],
[[3, 5, 2, 0], 1, 60, 0],
[[3, 5, 3, 9], -1, -60, 0],
[[3, 5, 4, 7], -1, -60, 0],
[[3, 5, 5, 5], 1, 60, 0],
[[3, 5, 6, 3], -1, -60, 0],
[[3, 5, 7, 1], 1, 60, 0],
[[3, 5, 8, 10], 0, 0, 0],
[[3, 5, 9, 8], -1, -60, 0],
[[3, 5, 10, 6], 0, 0, 0],
```

```
[[3, 6, 0, 4], 1, 60, 0],
[[3, 6, 1, 6], 0, 0, 0],
[[3, 6, 2, 8], -1, -60, 0],
[[3, 6, 3, 10], 0, 0, 0],
[[3, 6, 4, 1], 1, 60, 0],
[[3, 6, 5, 3], -1, -60, 0],
[[3, 6, 6, 5], 1, 60, 0],
[[3, 6, 7, 7], -1, -60, 0],
[[3, 6, 8, 9], -1, -60, 0],
[[3, 6, 9, 0], 1, 60, 0],
[[3, 6, 10, 2], -1, -60, 0],
[[3, 7, 0, 4], 1, 60, 0],
[[3, 7, 1, 10], 0, 0, 0],
[[3, 7, 2, 5], 1, 60, 0],
[[3, 7, 3, 0], 1, 60, 0],
[[3, 7, 4, 6], 0, 0, 0],
[[3, 7, 5, 1], 1, 60, 0],
[[3, 7, 6, 7], -1, -60, 0],
[[3, 7, 7, 2], -1, -60, 0],
[[3, 7, 8, 8], -1, -60, 0],
[[3, 7, 9, 3], -1, -60, 0],
[[3, 7, 10, 9], -1, -60, 0],
[[3, 8, 0, 4], 1, 60, 0],
[[3, 8, 1, 3], -1, -60, 0],
[[3, 8, 2, 2], -1, -60, 0],
[[3, 8, 3, 1], 1, 60, 0],
[[3, 8, 4, 0], 1, 60, 0],
[[3, 8, 5, 10], 0, 0, 0],
[[3, 8, 6, 9], -1, -60, 0],
[[3, 8, 7, 8], -1, -60, 0],
[[3, 8, 8, 7], -1, -60, 0],
[[3, 8, 9, 6], 0, 0, 0],
[[3, 8, 10, 5], 1, 60, 0],
[[3, 9, 0, 4], 1, 60, 0],
[[3, 9, 1, 7], -1, -60, 0],
[[3, 9, 2, 10], 0, 0, 0],
[[3, 9, 3, 2], -1, -60, 0],
[[3, 9, 4, 5], 1, 60, 0],
[[3, 9, 5, 8], -1, -60, 0],
[[3, 9, 6, 0], 1, 60, 0],
[[3, 9, 7, 3], -1, -60, 0],
[[3, 9, 8, 6], 0, 0, 0],
[[3, 9, 9, 9], -1, -60, 0],
[[3, 9, 10, 1], 1, 60, 0],
[[3, 10, 0, 4], 1, 60, 0],
[[3, 10, 1, 0], 1, 60, 0],
[[3, 10, 2, 7], -1, -60, 0],
[[3, 10, 3, 3], -1, -60, 0],
[[3, 10, 4, 10], 0, 0, 0],
[[3, 10, 5, 6], 0, 0, 0],
[[3, 10, 6, 2], -1, -60, 0],
[[3, 10, 7, 9], -1, -60, 0],
[[3, 10, 8, 5], 1, 60, 0],
[[3, 10, 9, 1], 1, 60, 0],
[[3, 10, 10, 8], -1, -60, 0],
[[4, 0, 0, 3], 1, 60, 0],
[[4, 0, 1, 3], 1, 60, 0],
[[4, 0, 2, 3], 1, 60, 0],
[[4, 0, 3, 3], 1, 60, 0],
```

```
[[4, 0, 4, 3], 1, 60, 0],
[[4, 0, 5, 3], 1, 60, 0],
[[4, 0, 6, 3], 1, 60, 0],
[[4, 0, 7, 3], 1, 60, 0],
[[4, 0, 8, 3], 1, 60, 0],
[[4, 0, 9, 3], 1, 60, 0],
[[4, 0, 10, 3], 1, 60, 0],
[[4, 1, 0, 3], 1, 60, 0],
[[4, 1, 1, 6], -1, -60, 0],
[[4, 1, 2, 9], 0, 0, 0],
[[4, 1, 3, 1], -1, -60, 0],
[[4, 1, 4, 4], 1, 60, 0],
[[4, 1, 5, 7], -1, -60, 0],
[[4, 1, 6, 10], 1, 60, 0],
[[4, 1, 7, 2], -1, -60, 0],
[[4, 1, 8, 5], 0, 0, 0],
[[4, 1, 9, 8], -1, -60, 0],
[[4, 1, 10, 0], 1, 60, 0],
[[4, 2, 0, 3], 1, 60, 0],
[[4, 2, 1, 9], 0, 0, 0],
[[4, 2, 2, 4], 1, 60, 0],
[[4, 2, 3, 10], 1, 60, 0],
[[4, 2, 4, 5], 0, 0, 0],
[[4, 2, 5, 0], 1, 60, 0],
[[4, 2, 6, 6], -1, -60, 0],
[[4, 2, 7, 1], -1, -60, 0],
[[4, 2, 8, 7], -1, -60, 0],
[[4, 2, 9, 2], -1, -60, 0],
[[4, 2, 10, 8], -1, -60, 0],
[[4, 3, 0, 3], 1, 60, 0],
[[4, 3, 1, 1], -1, -60, 0],
[[4, 3, 2, 10], 1, 60, 0],
[[4, 3, 3, 8], -1, -60, 0],
[[4, 3, 4, 6], -1, -60, 0],
[[4, 3, 5, 4], 1, 60, 0],
[[4, 3, 6, 2], -1, -60, 0],
[[4, 3, 7, 0], 1, 60, 0],
[[4, 3, 8, 9], 0, 0, 0],
[[4, 3, 9, 7], -1, -60, 0],
[[4, 3, 10, 5], 0, 0, 0],
[[4, 4, 0, 3], 1, 60, 0],
[[4, 4, 1, 4], 1, 60, 0],
[[4, 4, 2, 5], 0, 0, 0],
[[4, 4, 3, 6], -1, -60, 0],
[[4, 4, 4, 7], -1, -60, 0],
[[4, 4, 5, 8], -1, -60, 0],
[[4, 4, 6, 9], 0, 0, 0],
[[4, 4, 7, 10], 1, 60, 0],
[[4, 4, 8, 0], 1, 60, 0],
[[4, 4, 9, 1], -1, -60, 0],
[[4, 4, 10, 2], -1, -60, 0],
[[4, 5, 0, 3], 1, 60, 0],
[[4, 5, 1, 7], -1, -60, 0],
[[4, 5, 2, 0], 1, 60, 0],
[[4, 5, 3, 4], 1, 60, 0],
[[4, 5, 4, 8], -1, -60, 0],
[[4, 5, 5, 1], -1, -60, 0],
[[4, 5, 6, 5], 0, 0, 0],
[[4, 5, 7, 9], 0, 0, 0],
```

```
[[4, 5, 8, 2], -1, -60, 0],
[[4, 5, 9, 6], -1, -60, 0],
[[4, 5, 10, 10], 1, 60, 0],
[[4, 6, 0, 3], 1, 60, 0],
[[4, 6, 1, 10], 1, 60, 0],
[[4, 6, 2, 6], -1, -60, 0],
[[4, 6, 3, 2], -1, -60, 0],
[[4, 6, 4, 9], 0, 0, 0],
[[4, 6, 5, 5], 0, 0, 0],
[[4, 6, 6, 1], -1, -60, 0],
[[4, 6, 7, 8], -1, -60, 0],
[[4, 6, 8, 4], 1, 60, 0],
[[4, 6, 9, 0], 1, 60, 0],
[[4, 6, 10, 7], -1, -60, 0],
[[4, 7, 0, 3], 1, 60, 0],
[[4, 7, 1, 2], -1, -60, 0],
[[4, 7, 2, 1], -1, -60, 0],
[[4, 7, 3, 0], 1, 60, 0],
[[4, 7, 4, 10], 1, 60, 0],
[[4, 7, 5, 9], 0, 0, 0],
[[4, 7, 6, 8], -1, -60, 0],
[[4, 7, 7, 7], -1, -60, 0],
[[4, 7, 8, 6], -1, -60, 0],
[[4, 7, 9, 5], 0, 0, 0],
[[4, 7, 10, 4], 1, 60, 0],
[[4, 8, 0, 3], 1, 60, 0],
[[4, 8, 1, 5], 0, 0, 0],
[[4, 8, 2, 7], -1, -60, 0],
[[4, 8, 3, 9], 0, 0, 0],
[[4, 8, 4, 0], 1, 60, 0],
[[4, 8, 5, 2], -1, -60, 0],
[[4, 8, 6, 4], 1, 60, 0],
[[4, 8, 7, 6], -1, -60, 0],
[[4, 8, 8, 8], -1, -60, 0],
[[4, 8, 9, 10], 1, 60, 0],
[[4, 8, 10, 1], -1, -60, 0],
[[4, 9, 0, 3], 1, 60, 0],
[[4, 9, 1, 8], -1, -60, 0],
[[4, 9, 2, 2], -1, -60, 0],
[[4, 9, 3, 7], -1, -60, 0],
[[4, 9, 4, 1], -1, -60, 0],
[[4, 9, 5, 6], -1, -60, 0],
[[4, 9, 6, 0], 1, 60, 0],
[[4, 9, 7, 5], 0, 0, 0],
[[4, 9, 8, 10], 1, 60, 0],
[[4, 9, 9, 4], 1, 60, 0],
[[4, 9, 10, 9], 0, 0, 0],
[[4, 10, 0, 3], 1, 60, 0],
[[4, 10, 1, 0], 1, 60, 0],
[[4, 10, 2, 8], -1, -60, 0],
[[4, 10, 3, 5], 0, 0, 0],
[[4, 10, 4, 2], -1, -60, 0],
[[4, 10, 5, 10], 1, 60, 0],
[[4, 10, 6, 7], -1, -60, 0],
[[4, 10, 7, 4], 1, 60, 0],
[[4, 10, 8, 1], -1, -60, 0],
[[4, 10, 9, 9], 0, 0, 0],
[[4, 10, 10, 6], -1, -60, 0],
[[5, 0, 0, 9], 1, 60, 0],
```

```
[[5, 0, 1, 9], 1, 60, 0],
[[5, 0, 2, 9], 1, 60, 0],
[[5, 0, 3, 9], 1, 60, 0],
[[5, 0, 4, 9], 1, 60, 0],
[[5, 0, 5, 9], 1, 60, 0],
[[5, 0, 6, 9], 1, 60, 0],
[[5, 0, 7, 9], 1, 60, 0],
[[5, 0, 8, 9], 1, 60, 0],
[[5, 0, 9, 9], 1, 60, 0],
[[5, 0, 10, 9], 1, 60, 0],
[[5, 1, 0, 9], 1, 60, 0],
[[5, 1, 1, 7], -1, -60, 0],
[[5, 1, 2, 5], -1, -60, 0],
[[5, 1, 3, 3], 1, 60, 0],
[[5, 1, 4, 1], -1, -60, 0],
[[5, 1, 5, 10], 1, 60, 0],
[[5, 1, 6, 8], 0, 0, 0],
[[5, 1, 7, 6], -1, -60, 0],
[[5, 1, 8, 4], 0, 0, 0],
[[5, 1, 9, 2], 1, 60, 0],
[[5, 1, 10, 0], -1, -60, 0],
[[5, 2, 0, 9], 1, 60, 0],
[[5, 2, 1, 5], -1, -60, 0],
[[5, 2, 2, 1], -1, -60, 0],
[[5, 2, 3, 8], 0, 0, 0],
[[5, 2, 4, 4], 0, 0, 0],
[[5, 2, 5, 0], -1, -60, 0],
[[5, 2, 6, 7], -1, -60, 0],
[[5, 2, 7, 3], 1, 60, 0],
[[5, 2, 8, 10], 1, 60, 0],
[[5, 2, 9, 6], -1, -60, 0],
[[5, 2, 10, 2], 1, 60, 0],
[[5, 3, 0, 9], 1, 60, 0],
[[5, 3, 1, 3], 1, 60, 0],
[[5, 3, 2, 8], 0, 0, 0],
[[5, 3, 3, 2], 1, 60, 0],
[[5, 3, 4, 7], -1, -60, 0],
[[5, 3, 5, 1], -1, -60, 0],
[[5, 3, 6, 6], -1, -60, 0],
[[5, 3, 7, 0], -1, -60, 0],
[[5, 3, 8, 5], -1, -60, 0],
[[5, 3, 9, 10], 1, 60, 0],
[[5, 3, 10, 4], 0, 0, 0],
[[5, 4, 0, 9], 1, 60, 0],
[[5, 4, 1, 1], -1, -60, 0],
[[5, 4, 2, 4], 0, 0, 0],
[[5, 4, 3, 7], -1, -60, 0],
[[5, 4, 4, 10], 1, 60, 0],
[[5, 4, 5, 2], 1, 60, 0],
[[5, 4, 6, 5], -1, -60, 0],
[[5, 4, 7, 8], 0, 0, 0],
[[5, 4, 8, 0], -1, -60, 0],
[[5, 4, 9, 3], 1, 60, 0],
[[5, 4, 10, 6], -1, -60, 0],
[[5, 5, 0, 9], 1, 60, 0],
[[5, 5, 1, 10], 1, 60, 0],
[[5, 5, 2, 0], -1, -60, 0],
[[5, 5, 3, 1], -1, -60, 0],
[[5, 5, 4, 2], 1, 60, 0],
```

```
[[5, 5, 5, 3], 1, 60, 0],
[[5, 5, 6, 4], 0, 0, 0],
[[5, 5, 7, 5], -1, -60, 0],
[[5, 5, 8, 6], -1, -60, 0],
[[5, 5, 9, 7], -1, -60, 0],
[[5, 5, 10, 8], 0, 0, 0],
[[5, 6, 0, 9], 1, 60, 0],
[[5, 6, 1, 8], 0, 0, 0],
[[5, 6, 2, 7], -1, -60, 0],
[[5, 6, 3, 6], -1, -60, 0],
[[5, 6, 4, 5], -1, -60, 0],
[[5, 6, 5, 4], 0, 0, 0],
[[5, 6, 6, 3], 1, 60, 0],
[[5, 6, 7, 2], 1, 60, 0],
[[5, 6, 8, 1], -1, -60, 0],
[[5, 6, 9, 0], -1, -60, 0],
[[5, 6, 10, 10], 1, 60, 0],
[[5, 7, 0, 9], 1, 60, 0],
[[5, 7, 1, 6], -1, -60, 0],
[[5, 7, 2, 3], 1, 60, 0],
[[5, 7, 3, 0], -1, -60, 0],
[[5, 7, 4, 8], 0, 0, 0],
[[5, 7, 5, 5], -1, -60, 0],
[[5, 7, 6, 2], 1, 60, 0],
[[5, 7, 7, 10], 1, 60, 0],
[[5, 7, 8, 7], -1, -60, 0],
[[5, 7, 9, 4], 0, 0, 0],
[[5, 7, 10, 1], -1, -60, 0],
[[5, 8, 0, 9], 1, 60, 0],
[[5, 8, 1, 4], 0, 0, 0],
[[5, 8, 2, 10], 1, 60, 0],
[[5, 8, 3, 5], -1, -60, 0],
[[5, 8, 4, 0], -1, -60, 0],
[[5, 8, 5, 6], -1, -60, 0],
[[5, 8, 6, 1], -1, -60, 0],
[[5, 8, 7, 7], -1, -60, 0],
[[5, 8, 8, 2], 1, 60, 0],
[[5, 8, 9, 8], 0, 0, 0],
[[5, 8, 10, 3], 1, 60, 0],
[[5, 9, 0, 9], 1, 60, 0],
[[5, 9, 1, 2], 1, 60, 0],
[[5, 9, 2, 6], -1, -60, 0],
[[5, 9, 3, 10], 1, 60, 0],
[[5, 9, 4, 3], 1, 60, 0],
[[5, 9, 5, 7], -1, -60, 0],
[[5, 9, 6, 0], -1, -60, 0],
[[5, 9, 7, 4], 0, 0, 0],
[[5, 9, 8, 8], 0, 0, 0],
[[5, 9, 9, 1], -1, -60, 0],
[[5, 9, 10, 5], -1, -60, 0],
[[5, 10, 0, 9], 1, 60, 0],
[[5, 10, 1, 0], -1, -60, 0],
[[5, 10, 2, 2], 1, 60, 0],
[[5, 10, 3, 4], 0, 0, 0],
[[5, 10, 4, 6], -1, -60, 0],
[[5, 10, 5, 8], 0, 0, 0],
[[5, 10, 6, 10], 1, 60, 0],
[[5, 10, 7, 1], -1, -60, 0],
[[5, 10, 8, 3], 1, 60, 0],
```

```
        [[5, 10, 9, 5], -1, -60, 0],
        [[5, 10, 10, 7], -1, -60, 0]
      ]
    },
    {
      "p": 13,
      "group_order": 1092,
      "u": 84,
      "H_coefficient_distribution": {"-1": 468, "0": 168, "1": 455, "13": 1},
      "augmentation": 0,
      "H_squared_minus_uH_nonzero_entries": [],
      "number_of_residual_coefficients_checked": 1092,
      "coefficient_table_columns": ["matrix_a_b_c_d", "H", "H_squared", "H_squared_minus_uH
"],
      "complete_coefficient_table": [
        [[0, 1, 12, 0], 1, 84, 0],
        [[0, 1, 12, 1], 1, 84, 0],
        [[0, 1, 12, 2], 0, 0, 0],
        [[0, 1, 12, 3], -1, -84, 0],
        [[0, 1, 12, 4], 1, 84, 0],
        [[0, 1, 12, 5], -1, -84, 0],
        [[0, 1, 12, 6], -1, -84, 0],
        [[0, 1, 12, 7], -1, -84, 0],
        [[0, 1, 12, 8], -1, -84, 0],
        [[0, 1, 12, 9], 1, 84, 0],
        [[0, 1, 12, 10], -1, -84, 0],
        [[0, 1, 12, 11], 0, 0, 0],
        [[0, 1, 12, 12], 1, 84, 0],
        [[0, 2, 6, 0], 1, 84, 0],
        [[0, 2, 6, 1], 1, 84, 0],
        [[0, 2, 6, 2], 0, 0, 0],
        [[0, 2, 6, 3], -1, -84, 0],
        [[0, 2, 6, 4], 1, 84, 0],
        [[0, 2, 6, 5], -1, -84, 0],
        [[0, 2, 6, 6], -1, -84, 0],
        [[0, 2, 6, 7], -1, -84, 0],
        [[0, 2, 6, 8], -1, -84, 0],
        [[0, 2, 6, 9], 1, 84, 0],
        [[0, 2, 6, 10], -1, -84, 0],
        [[0, 2, 6, 11], 0, 0, 0],
        [[0, 2, 6, 12], 1, 84, 0],
        [[0, 3, 4, 0], 1, 84, 0],
        [[0, 3, 4, 1], 1, 84, 0],
        [[0, 3, 4, 2], 0, 0, 0],
        [[0, 3, 4, 3], -1, -84, 0],
        [[0, 3, 4, 4], 1, 84, 0],
        [[0, 3, 4, 5], -1, -84, 0],
        [[0, 3, 4, 6], -1, -84, 0],
        [[0, 3, 4, 7], -1, -84, 0],
        [[0, 3, 4, 8], -1, -84, 0],
        [[0, 3, 4, 9], 1, 84, 0],
        [[0, 3, 4, 10], -1, -84, 0],
        [[0, 3, 4, 11], 0, 0, 0],
        [[0, 3, 4, 12], 1, 84, 0],
        [[0, 4, 3, 0], 1, 84, 0],
        [[0, 4, 3, 1], 1, 84, 0],
        [[0, 4, 3, 2], 0, 0, 0],
        [[0, 4, 3, 3], -1, -84, 0],
        [[0, 4, 3, 4], 1, 84, 0],
```

```
[[0, 4, 3, 5], -1, -84, 0],
[[0, 4, 3, 6], -1, -84, 0],
[[0, 4, 3, 7], -1, -84, 0],
[[0, 4, 3, 8], -1, -84, 0],
[[0, 4, 3, 9], 1, 84, 0],
[[0, 4, 3, 10], -1, -84, 0],
[[0, 4, 3, 11], 0, 0, 0],
[[0, 4, 3, 12], 1, 84, 0],
[[0, 5, 5, 0], 1, 84, 0],
[[0, 5, 5, 1], 1, 84, 0],
[[0, 5, 5, 2], 0, 0, 0],
[[0, 5, 5, 3], -1, -84, 0],
[[0, 5, 5, 4], 1, 84, 0],
[[0, 5, 5, 5], -1, -84, 0],
[[0, 5, 5, 6], -1, -84, 0],
[[0, 5, 5, 7], -1, -84, 0],
[[0, 5, 5, 8], -1, -84, 0],
[[0, 5, 5, 9], 1, 84, 0],
[[0, 5, 5, 10], -1, -84, 0],
[[0, 5, 5, 11], 0, 0, 0],
[[0, 5, 5, 12], 1, 84, 0],
[[0, 6, 2, 0], 1, 84, 0],
[[0, 6, 2, 1], 1, 84, 0],
[[0, 6, 2, 2], 0, 0, 0],
[[0, 6, 2, 3], -1, -84, 0],
[[0, 6, 2, 4], 1, 84, 0],
[[0, 6, 2, 5], -1, -84, 0],
[[0, 6, 2, 6], -1, -84, 0],
[[0, 6, 2, 7], -1, -84, 0],
[[0, 6, 2, 8], -1, -84, 0],
[[0, 6, 2, 9], 1, 84, 0],
[[0, 6, 2, 10], -1, -84, 0],
[[0, 6, 2, 11], 0, 0, 0],
[[0, 6, 2, 12], 1, 84, 0],
[[1, 0, 0, 1], 13, 1092, 0],
[[1, 0, 1, 1], 0, 0, 0],
[[1, 0, 2, 1], 0, 0, 0],
[[1, 0, 3, 1], 0, 0, 0],
[[1, 0, 4, 1], 0, 0, 0],
[[1, 0, 5, 1], 0, 0, 0],
[[1, 0, 6, 1], 0, 0, 0],
[[1, 0, 7, 1], 0, 0, 0],
[[1, 0, 8, 1], 0, 0, 0],
[[1, 0, 9, 1], 0, 0, 0],
[[1, 0, 10, 1], 0, 0, 0],
[[1, 0, 11, 1], 0, 0, 0],
[[1, 0, 12, 1], 0, 0, 0],
[[1, 1, 0, 1], 0, 0, 0],
[[1, 1, 1, 2], -1, -84, 0],
[[1, 1, 2, 3], 1, 84, 0],
[[1, 1, 3, 4], -1, -84, 0],
[[1, 1, 4, 5], -1, -84, 0],
[[1, 1, 5, 6], -1, -84, 0],
[[1, 1, 6, 7], -1, -84, 0],
[[1, 1, 7, 8], 1, 84, 0],
[[1, 1, 8, 9], -1, -84, 0],
[[1, 1, 9, 10], 0, 0, 0],
[[1, 1, 10, 11], 1, 84, 0],
[[1, 1, 11, 12], 1, 84, 0],
```

```
[[1, 1, 12, 0], 1, 84, 0],
[[1, 2, 0, 1], 0, 0, 0],
[[1, 2, 1, 3], 1, 84, 0],
[[1, 2, 2, 5], -1, -84, 0],
[[1, 2, 3, 7], -1, -84, 0],
[[1, 2, 4, 9], -1, -84, 0],
[[1, 2, 5, 11], 1, 84, 0],
[[1, 2, 6, 0], 1, 84, 0],
[[1, 2, 7, 2], -1, -84, 0],
[[1, 2, 8, 4], -1, -84, 0],
[[1, 2, 9, 6], -1, -84, 0],
[[1, 2, 10, 8], 1, 84, 0],
[[1, 2, 11, 10], 0, 0, 0],
[[1, 2, 12, 12], 1, 84, 0],
[[1, 3, 0, 1], 0, 0, 0],
[[1, 3, 1, 4], -1, -84, 0],
[[1, 3, 2, 7], -1, -84, 0],
[[1, 3, 3, 10], 0, 0, 0],
[[1, 3, 4, 0], 1, 84, 0],
[[1, 3, 5, 3], 1, 84, 0],
[[1, 3, 6, 6], -1, -84, 0],
[[1, 3, 7, 9], -1, -84, 0],
[[1, 3, 8, 12], 1, 84, 0],
[[1, 3, 9, 2], -1, -84, 0],
[[1, 3, 10, 5], -1, -84, 0],
[[1, 3, 11, 8], 1, 84, 0],
[[1, 3, 12, 11], 1, 84, 0],
[[1, 4, 0, 1], 0, 0, 0],
[[1, 4, 1, 5], -1, -84, 0],
[[1, 4, 2, 9], -1, -84, 0],
[[1, 4, 3, 0], 1, 84, 0],
[[1, 4, 4, 4], -1, -84, 0],
[[1, 4, 5, 8], 1, 84, 0],
[[1, 4, 6, 12], 1, 84, 0],
[[1, 4, 7, 3], 1, 84, 0],
[[1, 4, 8, 7], -1, -84, 0],
[[1, 4, 9, 11], 1, 84, 0],
[[1, 4, 10, 2], -1, -84, 0],
[[1, 4, 11, 6], -1, -84, 0],
[[1, 4, 12, 10], 0, 0, 0],
[[1, 5, 0, 1], 0, 0, 0],
[[1, 5, 1, 6], -1, -84, 0],
[[1, 5, 2, 11], 1, 84, 0],
[[1, 5, 3, 3], 1, 84, 0],
[[1, 5, 4, 8], 1, 84, 0],
[[1, 5, 5, 0], 1, 84, 0],
[[1, 5, 6, 5], -1, -84, 0],
[[1, 5, 7, 10], 0, 0, 0],
[[1, 5, 8, 2], -1, -84, 0],
[[1, 5, 9, 7], -1, -84, 0],
[[1, 5, 10, 12], 1, 84, 0],
[[1, 5, 11, 4], -1, -84, 0],
[[1, 5, 12, 9], -1, -84, 0],
[[1, 6, 0, 1], 0, 0, 0],
[[1, 6, 1, 7], -1, -84, 0],
[[1, 6, 2, 0], 1, 84, 0],
[[1, 6, 3, 6], -1, -84, 0],
[[1, 6, 4, 12], 1, 84, 0],
[[1, 6, 5, 5], -1, -84, 0],
```

```
[[1, 6, 6, 11], 1, 84, 0],
[[1, 6, 7, 4], -1, -84, 0],
[[1, 6, 8, 10], 0, 0, 0],
[[1, 6, 9, 3], 1, 84, 0],
[[1, 6, 10, 9], -1, -84, 0],
[[1, 6, 11, 2], -1, -84, 0],
[[1, 6, 12, 8], 1, 84, 0],
[[1, 7, 0, 1], 0, 0, 0],
[[1, 7, 1, 8], 1, 84, 0],
[[1, 7, 2, 2], -1, -84, 0],
[[1, 7, 3, 9], -1, -84, 0],
[[1, 7, 4, 3], 1, 84, 0],
[[1, 7, 5, 10], 0, 0, 0],
[[1, 7, 6, 4], -1, -84, 0],
[[1, 7, 7, 11], 1, 84, 0],
[[1, 7, 8, 5], -1, -84, 0],
[[1, 7, 9, 12], 1, 84, 0],
[[1, 7, 10, 6], -1, -84, 0],
[[1, 7, 11, 0], 1, 84, 0],
[[1, 7, 12, 7], -1, -84, 0],
[[1, 8, 0, 1], 0, 0, 0],
[[1, 8, 1, 9], -1, -84, 0],
[[1, 8, 2, 4], -1, -84, 0],
[[1, 8, 3, 12], 1, 84, 0],
[[1, 8, 4, 7], -1, -84, 0],
[[1, 8, 5, 2], -1, -84, 0],
[[1, 8, 6, 10], 0, 0, 0],
[[1, 8, 7, 5], -1, -84, 0],
[[1, 8, 8, 0], 1, 84, 0],
[[1, 8, 9, 8], 1, 84, 0],
[[1, 8, 10, 3], 1, 84, 0],
[[1, 8, 11, 11], 1, 84, 0],
[[1, 8, 12, 6], -1, -84, 0],
[[1, 9, 0, 1], 0, 0, 0],
[[1, 9, 1, 10], 0, 0, 0],
[[1, 9, 2, 6], -1, -84, 0],
[[1, 9, 3, 2], -1, -84, 0],
[[1, 9, 4, 11], 1, 84, 0],
[[1, 9, 5, 7], -1, -84, 0],
[[1, 9, 6, 3], 1, 84, 0],
[[1, 9, 7, 12], 1, 84, 0],
[[1, 9, 8, 8], 1, 84, 0],
[[1, 9, 9, 4], -1, -84, 0],
[[1, 9, 10, 0], 1, 84, 0],
[[1, 9, 11, 9], -1, -84, 0],
[[1, 9, 12, 5], -1, -84, 0],
[[1, 10, 0, 1], 0, 0, 0],
[[1, 10, 1, 11], 1, 84, 0],
[[1, 10, 2, 8], 1, 84, 0],
[[1, 10, 3, 5], -1, -84, 0],
[[1, 10, 4, 2], -1, -84, 0],
[[1, 10, 5, 12], 1, 84, 0],
[[1, 10, 6, 9], -1, -84, 0],
[[1, 10, 7, 6], -1, -84, 0],
[[1, 10, 8, 3], 1, 84, 0],
[[1, 10, 9, 0], 1, 84, 0],
[[1, 10, 10, 10], 0, 0, 0],
[[1, 10, 11, 7], -1, -84, 0],
[[1, 10, 12, 4], -1, -84, 0],
```

```
[[1, 11, 0, 1], 0, 0, 0],
[[1, 11, 1, 12], 1, 84, 0],
[[1, 11, 2, 10], 0, 0, 0],
[[1, 11, 3, 8], 1, 84, 0],
[[1, 11, 4, 6], -1, -84, 0],
[[1, 11, 5, 4], -1, -84, 0],
[[1, 11, 6, 2], -1, -84, 0],
[[1, 11, 7, 0], 1, 84, 0],
[[1, 11, 8, 11], 1, 84, 0],
[[1, 11, 9, 9], -1, -84, 0],
[[1, 11, 10, 7], -1, -84, 0],
[[1, 11, 11, 5], -1, -84, 0],
[[1, 11, 12, 3], 1, 84, 0],
[[1, 12, 0, 1], 0, 0, 0],
[[1, 12, 1, 0], 1, 84, 0],
[[1, 12, 2, 12], 1, 84, 0],
[[1, 12, 3, 11], 1, 84, 0],
[[1, 12, 4, 10], 0, 0, 0],
[[1, 12, 5, 9], -1, -84, 0],
[[1, 12, 6, 8], 1, 84, 0],
[[1, 12, 7, 7], -1, -84, 0],
[[1, 12, 8, 6], -1, -84, 0],
[[1, 12, 9, 5], -1, -84, 0],
[[1, 12, 10, 4], -1, -84, 0],
[[1, 12, 11, 3], 1, 84, 0],
[[1, 12, 12, 2], -1, -84, 0],
[[2, 0, 0, 7], 1, 84, 0],
[[2, 0, 1, 7], 1, 84, 0],
[[2, 0, 2, 7], 1, 84, 0],
[[2, 0, 3, 7], 1, 84, 0],
[[2, 0, 4, 7], 1, 84, 0],
[[2, 0, 5, 7], 1, 84, 0],
[[2, 0, 6, 7], 1, 84, 0],
[[2, 0, 7, 7], 1, 84, 0],
[[2, 0, 8, 7], 1, 84, 0],
[[2, 0, 9, 7], 1, 84, 0],
[[2, 0, 10, 7], 1, 84, 0],
[[2, 0, 11, 7], 1, 84, 0],
[[2, 0, 12, 7], 1, 84, 0],
[[2, 1, 0, 7], 1, 84, 0],
[[2, 1, 1, 1], -1, -84, 0],
[[2, 1, 2, 8], -1, -84, 0],
[[2, 1, 3, 2], 1, 84, 0],
[[2, 1, 4, 9], 0, 0, 0],
[[2, 1, 5, 3], -1, -84, 0],
[[2, 1, 6, 10], 1, 84, 0],
[[2, 1, 7, 4], -1, -84, 0],
[[2, 1, 8, 11], 1, 84, 0],
[[2, 1, 9, 5], -1, -84, 0],
[[2, 1, 10, 12], 1, 84, 0],
[[2, 1, 11, 6], -1, -84, 0],
[[2, 1, 12, 0], 0, 0, 0],
[[2, 2, 0, 7], 1, 84, 0],
[[2, 2, 1, 8], -1, -84, 0],
[[2, 2, 2, 9], 0, 0, 0],
[[2, 2, 3, 10], 1, 84, 0],
[[2, 2, 4, 11], 1, 84, 0],
[[2, 2, 5, 12], 1, 84, 0],
[[2, 2, 6, 0], 0, 0, 0],
```

```
[[2, 2, 7, 1], -1, -84, 0],
[[2, 2, 8, 2], 1, 84, 0],
[[2, 2, 9, 3], -1, -84, 0],
[[2, 2, 10, 4], -1, -84, 0],
[[2, 2, 11, 5], -1, -84, 0],
[[2, 2, 12, 6], -1, -84, 0],
[[2, 3, 0, 7], 1, 84, 0],
[[2, 3, 1, 2], 1, 84, 0],
[[2, 3, 2, 10], 1, 84, 0],
[[2, 3, 3, 5], -1, -84, 0],
[[2, 3, 4, 0], 0, 0, 0],
[[2, 3, 5, 8], -1, -84, 0],
[[2, 3, 6, 3], -1, -84, 0],
[[2, 3, 7, 11], 1, 84, 0],
[[2, 3, 8, 6], -1, -84, 0],
[[2, 3, 9, 1], -1, -84, 0],
[[2, 3, 10, 9], 0, 0, 0],
[[2, 3, 11, 4], -1, -84, 0],
[[2, 3, 12, 12], 1, 84, 0],
[[2, 4, 0, 7], 1, 84, 0],
[[2, 4, 1, 9], 0, 0, 0],
[[2, 4, 2, 11], 1, 84, 0],
[[2, 4, 3, 0], 0, 0, 0],
[[2, 4, 4, 2], 1, 84, 0],
[[2, 4, 5, 4], -1, -84, 0],
[[2, 4, 6, 6], -1, -84, 0],
[[2, 4, 7, 8], -1, -84, 0],
[[2, 4, 8, 10], 1, 84, 0],
[[2, 4, 9, 12], 1, 84, 0],
[[2, 4, 10, 1], -1, -84, 0],
[[2, 4, 11, 3], -1, -84, 0],
[[2, 4, 12, 5], -1, -84, 0],
[[2, 5, 0, 7], 1, 84, 0],
[[2, 5, 1, 3], -1, -84, 0],
[[2, 5, 2, 12], 1, 84, 0],
[[2, 5, 3, 8], -1, -84, 0],
[[2, 5, 4, 4], -1, -84, 0],
[[2, 5, 5, 0], 0, 0, 0],
[[2, 5, 6, 9], 0, 0, 0],
[[2, 5, 7, 5], -1, -84, 0],
[[2, 5, 8, 1], -1, -84, 0],
[[2, 5, 9, 10], 1, 84, 0],
[[2, 5, 10, 6], -1, -84, 0],
[[2, 5, 11, 2], 1, 84, 0],
[[2, 5, 12, 11], 1, 84, 0],
[[2, 6, 0, 7], 1, 84, 0],
[[2, 6, 1, 10], 1, 84, 0],
[[2, 6, 2, 0], 0, 0, 0],
[[2, 6, 3, 3], -1, -84, 0],
[[2, 6, 4, 6], -1, -84, 0],
[[2, 6, 5, 9], 0, 0, 0],
[[2, 6, 6, 12], 1, 84, 0],
[[2, 6, 7, 2], 1, 84, 0],
[[2, 6, 8, 5], -1, -84, 0],
[[2, 6, 9, 8], -1, -84, 0],
[[2, 6, 10, 11], 1, 84, 0],
[[2, 6, 11, 1], -1, -84, 0],
[[2, 6, 12, 4], -1, -84, 0],
[[2, 7, 0, 7], 1, 84, 0],
```

```
[[2, 7, 1, 4], -1, -84, 0],
[[2, 7, 2, 1], -1, -84, 0],
[[2, 7, 3, 11], 1, 84, 0],
[[2, 7, 4, 8], -1, -84, 0],
[[2, 7, 5, 5], -1, -84, 0],
[[2, 7, 6, 2], 1, 84, 0],
[[2, 7, 7, 12], 1, 84, 0],
[[2, 7, 8, 9], 0, 0, 0],
[[2, 7, 9, 6], -1, -84, 0],
[[2, 7, 10, 3], -1, -84, 0],
[[2, 7, 11, 0], 0, 0, 0],
[[2, 7, 12, 10], 1, 84, 0],
[[2, 8, 0, 7], 1, 84, 0],
[[2, 8, 1, 11], 1, 84, 0],
[[2, 8, 2, 2], 1, 84, 0],
[[2, 8, 3, 6], -1, -84, 0],
[[2, 8, 4, 10], 1, 84, 0],
[[2, 8, 5, 1], -1, -84, 0],
[[2, 8, 6, 5], -1, -84, 0],
[[2, 8, 7, 9], 0, 0, 0],
[[2, 8, 8, 0], 0, 0, 0],
[[2, 8, 9, 4], -1, -84, 0],
[[2, 8, 10, 8], -1, -84, 0],
[[2, 8, 11, 12], 1, 84, 0],
[[2, 8, 12, 3], -1, -84, 0],
[[2, 9, 0, 7], 1, 84, 0],
[[2, 9, 1, 5], -1, -84, 0],
[[2, 9, 2, 3], -1, -84, 0],
[[2, 9, 3, 1], -1, -84, 0],
[[2, 9, 4, 12], 1, 84, 0],
[[2, 9, 5, 10], 1, 84, 0],
[[2, 9, 6, 8], -1, -84, 0],
[[2, 9, 7, 6], -1, -84, 0],
[[2, 9, 8, 4], -1, -84, 0],
[[2, 9, 9, 2], 1, 84, 0],
[[2, 9, 10, 0], 0, 0, 0],
[[2, 9, 11, 11], 1, 84, 0],
[[2, 9, 12, 9], 0, 0, 0],
[[2, 10, 0, 7], 1, 84, 0],
[[2, 10, 1, 12], 1, 84, 0],
[[2, 10, 2, 4], -1, -84, 0],
[[2, 10, 3, 9], 0, 0, 0],
[[2, 10, 4, 1], -1, -84, 0],
[[2, 10, 5, 6], -1, -84, 0],
[[2, 10, 6, 11], 1, 84, 0],
[[2, 10, 7, 3], -1, -84, 0],
[[2, 10, 8, 8], -1, -84, 0],
[[2, 10, 9, 0], 0, 0, 0],
[[2, 10, 10, 5], -1, -84, 0],
[[2, 10, 11, 10], 1, 84, 0],
[[2, 10, 12, 2], 1, 84, 0],
[[2, 11, 0, 7], 1, 84, 0],
[[2, 11, 1, 6], -1, -84, 0],
[[2, 11, 2, 5], -1, -84, 0],
[[2, 11, 3, 4], -1, -84, 0],
[[2, 11, 4, 3], -1, -84, 0],
[[2, 11, 5, 2], 1, 84, 0],
[[2, 11, 6, 1], -1, -84, 0],
[[2, 11, 7, 0], 0, 0, 0],
```

```
[[2, 11, 8, 12], 1, 84, 0],
[[2, 11, 9, 11], 1, 84, 0],
[[2, 11, 10, 10], 1, 84, 0],
[[2, 11, 11, 9], 0, 0, 0],
[[2, 11, 12, 8], -1, -84, 0],
[[2, 12, 0, 7], 1, 84, 0],
[[2, 12, 1, 0], 0, 0, 0],
[[2, 12, 2, 6], -1, -84, 0],
[[2, 12, 3, 12], 1, 84, 0],
[[2, 12, 4, 5], -1, -84, 0],
[[2, 12, 5, 11], 1, 84, 0],
[[2, 12, 6, 4], -1, -84, 0],
[[2, 12, 7, 10], 1, 84, 0],
[[2, 12, 8, 3], -1, -84, 0],
[[2, 12, 9, 9], 0, 0, 0],
[[2, 12, 10, 2], 1, 84, 0],
[[2, 12, 11, 8], -1, -84, 0],
[[2, 12, 12, 1], -1, -84, 0],
[[3, 0, 0, 9], 1, 84, 0],
[[3, 0, 1, 9], 1, 84, 0],
[[3, 0, 2, 9], 1, 84, 0],
[[3, 0, 3, 9], 1, 84, 0],
[[3, 0, 4, 9], 1, 84, 0],
[[3, 0, 5, 9], 1, 84, 0],
[[3, 0, 6, 9], 1, 84, 0],
[[3, 0, 7, 9], 1, 84, 0],
[[3, 0, 8, 9], 1, 84, 0],
[[3, 0, 9, 9], 1, 84, 0],
[[3, 0, 10, 9], 1, 84, 0],
[[3, 0, 11, 9], 1, 84, 0],
[[3, 0, 12, 9], 1, 84, 0],
[[3, 1, 0, 9], 1, 84, 0],
[[3, 1, 1, 5], -1, -84, 0],
[[3, 1, 2, 1], 1, 84, 0],
[[3, 1, 3, 10], 1, 84, 0],
[[3, 1, 4, 6], 1, 84, 0],
[[3, 1, 5, 2], -1, -84, 0],
[[3, 1, 6, 11], 1, 84, 0],
[[3, 1, 7, 7], -1, -84, 0],
[[3, 1, 8, 3], -1, -84, 0],
[[3, 1, 9, 12], 0, 0, 0],
[[3, 1, 10, 8], 0, 0, 0],
[[3, 1, 11, 4], -1, -84, 0],
[[3, 1, 12, 0], -1, -84, 0],
[[3, 2, 0, 9], 1, 84, 0],
[[3, 2, 1, 1], 1, 84, 0],
[[3, 2, 2, 6], 1, 84, 0],
[[3, 2, 3, 11], 1, 84, 0],
[[3, 2, 4, 3], -1, -84, 0],
[[3, 2, 5, 8], 0, 0, 0],
[[3, 2, 6, 0], -1, -84, 0],
[[3, 2, 7, 5], -1, -84, 0],
[[3, 2, 8, 10], 1, 84, 0],
[[3, 2, 9, 2], -1, -84, 0],
[[3, 2, 10, 7], -1, -84, 0],
[[3, 2, 11, 12], 0, 0, 0],
[[3, 2, 12, 4], -1, -84, 0],
[[3, 3, 0, 9], 1, 84, 0],
[[3, 3, 1, 10], 1, 84, 0],
```

```
[[3, 3, 2, 11], 1, 84, 0],
[[3, 3, 3, 12], 0, 0, 0],
[[3, 3, 4, 0], -1, -84, 0],
[[3, 3, 5, 1], 1, 84, 0],
[[3, 3, 6, 2], -1, -84, 0],
[[3, 3, 7, 3], -1, -84, 0],
[[3, 3, 8, 4], -1, -84, 0],
[[3, 3, 9, 5], -1, -84, 0],
[[3, 3, 10, 6], 1, 84, 0],
[[3, 3, 11, 7], -1, -84, 0],
[[3, 3, 12, 8], 0, 0, 0],
[[3, 4, 0, 9], 1, 84, 0],
[[3, 4, 1, 6], 1, 84, 0],
[[3, 4, 2, 3], -1, -84, 0],
[[3, 4, 3, 0], -1, -84, 0],
[[3, 4, 4, 10], 1, 84, 0],
[[3, 4, 5, 7], -1, -84, 0],
[[3, 4, 6, 4], -1, -84, 0],
[[3, 4, 7, 1], 1, 84, 0],
[[3, 4, 8, 11], 1, 84, 0],
[[3, 4, 9, 8], 0, 0, 0],
[[3, 4, 10, 5], -1, -84, 0],
[[3, 4, 11, 2], -1, -84, 0],
[[3, 4, 12, 12], 0, 0, 0],
[[3, 5, 0, 9], 1, 84, 0],
[[3, 5, 1, 2], -1, -84, 0],
[[3, 5, 2, 8], 0, 0, 0],
[[3, 5, 3, 1], 1, 84, 0],
[[3, 5, 4, 7], -1, -84, 0],
[[3, 5, 5, 0], -1, -84, 0],
[[3, 5, 6, 6], 1, 84, 0],
[[3, 5, 7, 12], 0, 0, 0],
[[3, 5, 8, 5], -1, -84, 0],
[[3, 5, 9, 11], 1, 84, 0],
[[3, 5, 10, 4], -1, -84, 0],
[[3, 5, 11, 10], 1, 84, 0],
[[3, 5, 12, 3], -1, -84, 0],
[[3, 6, 0, 9], 1, 84, 0],
[[3, 6, 1, 11], 1, 84, 0],
[[3, 6, 2, 0], -1, -84, 0],
[[3, 6, 3, 2], -1, -84, 0],
[[3, 6, 4, 4], -1, -84, 0],
[[3, 6, 5, 6], 1, 84, 0],
[[3, 6, 6, 8], 0, 0, 0],
[[3, 6, 7, 10], 1, 84, 0],
[[3, 6, 8, 12], 0, 0, 0],
[[3, 6, 9, 1], 1, 84, 0],
[[3, 6, 10, 3], -1, -84, 0],
[[3, 6, 11, 5], -1, -84, 0],
[[3, 6, 12, 7], -1, -84, 0],
[[3, 7, 0, 9], 1, 84, 0],
[[3, 7, 1, 7], -1, -84, 0],
[[3, 7, 2, 5], -1, -84, 0],
[[3, 7, 3, 3], -1, -84, 0],
[[3, 7, 4, 1], 1, 84, 0],
[[3, 7, 5, 12], 0, 0, 0],
[[3, 7, 6, 10], 1, 84, 0],
[[3, 7, 7, 8], 0, 0, 0],
[[3, 7, 8, 6], 1, 84, 0],
```

```
[[3, 7, 9, 4], -1, -84, 0],
[[3, 7, 10, 2], -1, -84, 0],
[[3, 7, 11, 0], -1, -84, 0],
[[3, 7, 12, 11], 1, 84, 0],
[[3, 8, 0, 9], 1, 84, 0],
[[3, 8, 1, 3], -1, -84, 0],
[[3, 8, 2, 10], 1, 84, 0],
[[3, 8, 3, 4], -1, -84, 0],
[[3, 8, 4, 11], 1, 84, 0],
[[3, 8, 5, 5], -1, -84, 0],
[[3, 8, 6, 12], 0, 0, 0],
[[3, 8, 7, 6], 1, 84, 0],
[[3, 8, 8, 0], -1, -84, 0],
[[3, 8, 9, 7], -1, -84, 0],
[[3, 8, 10, 1], 1, 84, 0],
[[3, 8, 11, 8], 0, 0, 0],
[[3, 8, 12, 2], -1, -84, 0],
[[3, 9, 0, 9], 1, 84, 0],
[[3, 9, 1, 12], 0, 0, 0],
[[3, 9, 2, 2], -1, -84, 0],
[[3, 9, 3, 5], -1, -84, 0],
[[3, 9, 4, 8], 0, 0, 0],
[[3, 9, 5, 11], 1, 84, 0],
[[3, 9, 6, 1], 1, 84, 0],
[[3, 9, 7, 4], -1, -84, 0],
[[3, 9, 8, 7], -1, -84, 0],
[[3, 9, 9, 10], 1, 84, 0],
[[3, 9, 10, 0], -1, -84, 0],
[[3, 9, 11, 3], -1, -84, 0],
[[3, 9, 12, 6], 1, 84, 0],
[[3, 10, 0, 9], 1, 84, 0],
[[3, 10, 1, 8], 0, 0, 0],
[[3, 10, 2, 7], -1, -84, 0],
[[3, 10, 3, 6], 1, 84, 0],
[[3, 10, 4, 5], -1, -84, 0],
[[3, 10, 5, 4], -1, -84, 0],
[[3, 10, 6, 3], -1, -84, 0],
[[3, 10, 7, 2], -1, -84, 0],
[[3, 10, 8, 1], 1, 84, 0],
[[3, 10, 9, 0], -1, -84, 0],
[[3, 10, 10, 12], 0, 0, 0],
[[3, 10, 11, 11], 1, 84, 0],
[[3, 10, 12, 10], 1, 84, 0],
[[3, 11, 0, 9], 1, 84, 0],
[[3, 11, 1, 4], -1, -84, 0],
[[3, 11, 2, 12], 0, 0, 0],
[[3, 11, 3, 7], -1, -84, 0],
[[3, 11, 4, 2], -1, -84, 0],
[[3, 11, 5, 10], 1, 84, 0],
[[3, 11, 6, 5], -1, -84, 0],
[[3, 11, 7, 0], -1, -84, 0],
[[3, 11, 8, 8], 0, 0, 0],
[[3, 11, 9, 3], -1, -84, 0],
[[3, 11, 10, 11], 1, 84, 0],
[[3, 11, 11, 6], 1, 84, 0],
[[3, 11, 12, 1], 1, 84, 0],
[[3, 12, 0, 9], 1, 84, 0],
[[3, 12, 1, 0], -1, -84, 0],
[[3, 12, 2, 4], -1, -84, 0],
```

```
[[3, 12, 3, 8], 0, 0, 0],
[[3, 12, 4, 12], 0, 0, 0],
[[3, 12, 5, 3], -1, -84, 0],
[[3, 12, 6, 7], -1, -84, 0],
[[3, 12, 7, 11], 1, 84, 0],
[[3, 12, 8, 2], -1, -84, 0],
[[3, 12, 9, 6], 1, 84, 0],
[[3, 12, 10, 10], 1, 84, 0],
[[3, 12, 11, 1], 1, 84, 0],
[[3, 12, 12, 5], -1, -84, 0],
[[4, 0, 0, 10], 1, 84, 0],
[[4, 0, 1, 10], 1, 84, 0],
[[4, 0, 2, 10], 1, 84, 0],
[[4, 0, 3, 10], 1, 84, 0],
[[4, 0, 4, 10], 1, 84, 0],
[[4, 0, 5, 10], 1, 84, 0],
[[4, 0, 6, 10], 1, 84, 0],
[[4, 0, 7, 10], 1, 84, 0],
[[4, 0, 8, 10], 1, 84, 0],
[[4, 0, 9, 10], 1, 84, 0],
[[4, 0, 10, 10], 1, 84, 0],
[[4, 0, 11, 10], 1, 84, 0],
[[4, 0, 12, 10], 1, 84, 0],
[[4, 1, 0, 10], 1, 84, 0],
[[4, 1, 1, 7], 0, 0, 0],
[[4, 1, 2, 4], -1, -84, 0],
[[4, 1, 3, 1], -1, -84, 0],
[[4, 1, 4, 11], 0, 0, 0],
[[4, 1, 5, 8], 1, 84, 0],
[[4, 1, 6, 5], 1, 84, 0],
[[4, 1, 7, 2], -1, -84, 0],
[[4, 1, 8, 12], -1, -84, 0],
[[4, 1, 9, 9], 1, 84, 0],
[[4, 1, 10, 6], -1, -84, 0],
[[4, 1, 11, 3], -1, -84, 0],
[[4, 1, 12, 0], 1, 84, 0],
[[4, 2, 0, 10], 1, 84, 0],
[[4, 2, 1, 4], -1, -84, 0],
[[4, 2, 2, 11], 0, 0, 0],
[[4, 2, 3, 5], 1, 84, 0],
[[4, 2, 4, 12], -1, -84, 0],
[[4, 2, 5, 6], -1, -84, 0],
[[4, 2, 6, 0], 1, 84, 0],
[[4, 2, 7, 7], 0, 0, 0],
[[4, 2, 8, 1], -1, -84, 0],
[[4, 2, 9, 8], 1, 84, 0],
[[4, 2, 10, 2], -1, -84, 0],
[[4, 2, 11, 9], 1, 84, 0],
[[4, 2, 12, 3], -1, -84, 0],
[[4, 3, 0, 10], 1, 84, 0],
[[4, 3, 1, 1], -1, -84, 0],
[[4, 3, 2, 5], 1, 84, 0],
[[4, 3, 3, 9], 1, 84, 0],
[[4, 3, 4, 0], 1, 84, 0],
[[4, 3, 5, 4], -1, -84, 0],
[[4, 3, 6, 8], 1, 84, 0],
[[4, 3, 7, 12], -1, -84, 0],
[[4, 3, 8, 3], -1, -84, 0],
[[4, 3, 9, 7], 0, 0, 0],
```

```
[[4, 3, 10, 11], 0, 0, 0],
[[4, 3, 11, 2], -1, -84, 0],
[[4, 3, 12, 6], -1, -84, 0],
[[4, 4, 0, 10], 1, 84, 0],
[[4, 4, 1, 11], 0, 0, 0],
[[4, 4, 2, 12], -1, -84, 0],
[[4, 4, 3, 0], 1, 84, 0],
[[4, 4, 4, 1], -1, -84, 0],
[[4, 4, 5, 2], -1, -84, 0],
[[4, 4, 6, 3], -1, -84, 0],
[[4, 4, 7, 4], -1, -84, 0],
[[4, 4, 8, 5], 1, 84, 0],
[[4, 4, 9, 6], -1, -84, 0],
[[4, 4, 10, 7], 0, 0, 0],
[[4, 4, 11, 8], 1, 84, 0],
[[4, 4, 12, 9], 1, 84, 0],
[[4, 5, 0, 10], 1, 84, 0],
[[4, 5, 1, 8], 1, 84, 0],
[[4, 5, 2, 6], -1, -84, 0],
[[4, 5, 3, 4], -1, -84, 0],
[[4, 5, 4, 2], -1, -84, 0],
[[4, 5, 5, 0], 1, 84, 0],
[[4, 5, 6, 11], 0, 0, 0],
[[4, 5, 7, 9], 1, 84, 0],
[[4, 5, 8, 7], 0, 0, 0],
[[4, 5, 9, 5], 1, 84, 0],
[[4, 5, 10, 3], -1, -84, 0],
[[4, 5, 11, 1], -1, -84, 0],
[[4, 5, 12, 12], -1, -84, 0],
[[4, 6, 0, 10], 1, 84, 0],
[[4, 6, 1, 5], 1, 84, 0],
[[4, 6, 2, 0], 1, 84, 0],
[[4, 6, 3, 8], 1, 84, 0],
[[4, 6, 4, 3], -1, -84, 0],
[[4, 6, 5, 11], 0, 0, 0],
[[4, 6, 6, 6], -1, -84, 0],
[[4, 6, 7, 1], -1, -84, 0],
[[4, 6, 8, 9], 1, 84, 0],
[[4, 6, 9, 4], -1, -84, 0],
[[4, 6, 10, 12], -1, -84, 0],
[[4, 6, 11, 7], 0, 0, 0],
[[4, 6, 12, 2], -1, -84, 0],
[[4, 7, 0, 10], 1, 84, 0],
[[4, 7, 1, 2], -1, -84, 0],
[[4, 7, 2, 7], 0, 0, 0],
[[4, 7, 3, 12], -1, -84, 0],
[[4, 7, 4, 4], -1, -84, 0],
[[4, 7, 5, 9], 1, 84, 0],
[[4, 7, 6, 1], -1, -84, 0],
[[4, 7, 7, 6], -1, -84, 0],
[[4, 7, 8, 11], 0, 0, 0],
[[4, 7, 9, 3], -1, -84, 0],
[[4, 7, 10, 8], 1, 84, 0],
[[4, 7, 11, 0], 1, 84, 0],
[[4, 7, 12, 5], 1, 84, 0],
[[4, 8, 0, 10], 1, 84, 0],
[[4, 8, 1, 12], -1, -84, 0],
[[4, 8, 2, 1], -1, -84, 0],
[[4, 8, 3, 3], -1, -84, 0],
```

```
[[4, 8, 4, 5], 1, 84, 0],
[[4, 8, 5, 7], 0, 0, 0],
[[4, 8, 6, 9], 1, 84, 0],
[[4, 8, 7, 11], 0, 0, 0],
[[4, 8, 8, 0], 1, 84, 0],
[[4, 8, 9, 2], -1, -84, 0],
[[4, 8, 10, 4], -1, -84, 0],
[[4, 8, 11, 6], -1, -84, 0],
[[4, 8, 12, 8], 1, 84, 0],
[[4, 9, 0, 10], 1, 84, 0],
[[4, 9, 1, 9], 1, 84, 0],
[[4, 9, 2, 8], 1, 84, 0],
[[4, 9, 3, 7], 0, 0, 0],
[[4, 9, 4, 6], -1, -84, 0],
[[4, 9, 5, 5], 1, 84, 0],
[[4, 9, 6, 4], -1, -84, 0],
[[4, 9, 7, 3], -1, -84, 0],
[[4, 9, 8, 2], -1, -84, 0],
[[4, 9, 9, 1], -1, -84, 0],
[[4, 9, 10, 0], 1, 84, 0],
[[4, 9, 11, 12], -1, -84, 0],
[[4, 9, 12, 11], 0, 0, 0],
[[4, 10, 0, 10], 1, 84, 0],
[[4, 10, 1, 6], -1, -84, 0],
[[4, 10, 2, 2], -1, -84, 0],
[[4, 10, 3, 11], 0, 0, 0],
[[4, 10, 4, 7], 0, 0, 0],
[[4, 10, 5, 3], -1, -84, 0],
[[4, 10, 6, 12], -1, -84, 0],
[[4, 10, 7, 8], 1, 84, 0],
[[4, 10, 8, 4], -1, -84, 0],
[[4, 10, 9, 0], 1, 84, 0],
[[4, 10, 10, 9], 1, 84, 0],
[[4, 10, 11, 5], 1, 84, 0],
[[4, 10, 12, 1], -1, -84, 0],
[[4, 11, 0, 10], 1, 84, 0],
[[4, 11, 1, 3], -1, -84, 0],
[[4, 11, 2, 9], 1, 84, 0],
[[4, 11, 3, 2], -1, -84, 0],
[[4, 11, 4, 8], 1, 84, 0],
[[4, 11, 5, 1], -1, -84, 0],
[[4, 11, 6, 7], 0, 0, 0],
[[4, 11, 7, 0], 1, 84, 0],
[[4, 11, 8, 6], -1, -84, 0],
[[4, 11, 9, 12], -1, -84, 0],
[[4, 11, 10, 5], 1, 84, 0],
[[4, 11, 11, 11], 0, 0, 0],
[[4, 11, 12, 4], -1, -84, 0],
[[4, 12, 0, 10], 1, 84, 0],
[[4, 12, 1, 0], 1, 84, 0],
[[4, 12, 2, 3], -1, -84, 0],
[[4, 12, 3, 6], -1, -84, 0],
[[4, 12, 4, 9], 1, 84, 0],
[[4, 12, 5, 12], -1, -84, 0],
[[4, 12, 6, 2], -1, -84, 0],
[[4, 12, 7, 5], 1, 84, 0],
[[4, 12, 8, 8], 1, 84, 0],
[[4, 12, 9, 11], 0, 0, 0],
[[4, 12, 10, 1], -1, -84, 0],
```

```
[[4, 12, 11, 4], -1, -84, 0],
[[4, 12, 12, 7], 0, 0, 0],
[[5, 0, 0, 8], 1, 84, 0],
[[5, 0, 1, 8], 1, 84, 0],
[[5, 0, 2, 8], 1, 84, 0],
[[5, 0, 3, 8], 1, 84, 0],
[[5, 0, 4, 8], 1, 84, 0],
[[5, 0, 5, 8], 1, 84, 0],
[[5, 0, 6, 8], 1, 84, 0],
[[5, 0, 7, 8], 1, 84, 0],
[[5, 0, 8, 8], 1, 84, 0],
[[5, 0, 9, 8], 1, 84, 0],
[[5, 0, 10, 8], 1, 84, 0],
[[5, 0, 11, 8], 1, 84, 0],
[[5, 0, 12, 8], 1, 84, 0],
[[5, 1, 0, 8], 1, 84, 0],
[[5, 1, 1, 3], -1, -84, 0],
[[5, 1, 2, 11], -1, -84, 0],
[[5, 1, 3, 6], 0, 0, 0],
[[5, 1, 4, 1], -1, -84, 0],
[[5, 1, 5, 9], 1, 84, 0],
[[5, 1, 6, 4], 1, 84, 0],
[[5, 1, 7, 12], 1, 84, 0],
[[5, 1, 8, 7], 1, 84, 0],
[[5, 1, 9, 2], -1, -84, 0],
[[5, 1, 10, 10], 0, 0, 0],
[[5, 1, 11, 5], -1, -84, 0],
[[5, 1, 12, 0], -1, -84, 0],
[[5, 2, 0, 8], 1, 84, 0],
[[5, 2, 1, 11], -1, -84, 0],
[[5, 2, 2, 1], -1, -84, 0],
[[5, 2, 3, 4], 1, 84, 0],
[[5, 2, 4, 7], 1, 84, 0],
[[5, 2, 5, 10], 0, 0, 0],
[[5, 2, 6, 0], -1, -84, 0],
[[5, 2, 7, 3], -1, -84, 0],
[[5, 2, 8, 6], 0, 0, 0],
[[5, 2, 9, 9], 1, 84, 0],
[[5, 2, 10, 12], 1, 84, 0],
[[5, 2, 11, 2], -1, -84, 0],
[[5, 2, 12, 5], -1, -84, 0],
[[5, 3, 0, 8], 1, 84, 0],
[[5, 3, 1, 6], 0, 0, 0],
[[5, 3, 2, 4], 1, 84, 0],
[[5, 3, 3, 2], -1, -84, 0],
[[5, 3, 4, 0], -1, -84, 0],
[[5, 3, 5, 11], -1, -84, 0],
[[5, 3, 6, 9], 1, 84, 0],
[[5, 3, 7, 7], 1, 84, 0],
[[5, 3, 8, 5], -1, -84, 0],
[[5, 3, 9, 3], -1, -84, 0],
[[5, 3, 10, 1], -1, -84, 0],
[[5, 3, 11, 12], 1, 84, 0],
[[5, 3, 12, 10], 0, 0, 0],
[[5, 4, 0, 8], 1, 84, 0],
[[5, 4, 1, 1], -1, -84, 0],
[[5, 4, 2, 7], 1, 84, 0],
[[5, 4, 3, 0], -1, -84, 0],
[[5, 4, 4, 6], 0, 0, 0],
```

```
[[5, 4, 5, 12], 1, 84, 0],
[[5, 4, 6, 5], -1, -84, 0],
[[5, 4, 7, 11], -1, -84, 0],
[[5, 4, 8, 4], 1, 84, 0],
[[5, 4, 9, 10], 0, 0, 0],
[[5, 4, 10, 3], -1, -84, 0],
[[5, 4, 11, 9], 1, 84, 0],
[[5, 4, 12, 2], -1, -84, 0],
[[5, 5, 0, 8], 1, 84, 0],
[[5, 5, 1, 9], 1, 84, 0],
[[5, 5, 2, 10], 0, 0, 0],
[[5, 5, 3, 11], -1, -84, 0],
[[5, 5, 4, 12], 1, 84, 0],
[[5, 5, 5, 0], -1, -84, 0],
[[5, 5, 6, 1], -1, -84, 0],
[[5, 5, 7, 2], -1, -84, 0],
[[5, 5, 8, 3], -1, -84, 0],
[[5, 5, 9, 4], 1, 84, 0],
[[5, 5, 10, 5], -1, -84, 0],
[[5, 5, 11, 6], 0, 0, 0],
[[5, 5, 12, 7], 1, 84, 0],
[[5, 6, 0, 8], 1, 84, 0],
[[5, 6, 1, 4], 1, 84, 0],
[[5, 6, 2, 0], -1, -84, 0],
[[5, 6, 3, 9], 1, 84, 0],
[[5, 6, 4, 5], -1, -84, 0],
[[5, 6, 5, 1], -1, -84, 0],
[[5, 6, 6, 10], 0, 0, 0],
[[5, 6, 7, 6], 0, 0, 0],
[[5, 6, 8, 2], -1, -84, 0],
[[5, 6, 9, 11], -1, -84, 0],
[[5, 6, 10, 7], 1, 84, 0],
[[5, 6, 11, 3], -1, -84, 0],
[[5, 6, 12, 12], 1, 84, 0],
[[5, 7, 0, 8], 1, 84, 0],
[[5, 7, 1, 12], 1, 84, 0],
[[5, 7, 2, 3], -1, -84, 0],
[[5, 7, 3, 7], 1, 84, 0],
[[5, 7, 4, 11], -1, -84, 0],
[[5, 7, 5, 2], -1, -84, 0],
[[5, 7, 6, 6], 0, 0, 0],
[[5, 7, 7, 10], 0, 0, 0],
[[5, 7, 8, 1], -1, -84, 0],
[[5, 7, 9, 5], -1, -84, 0],
[[5, 7, 10, 9], 1, 84, 0],
[[5, 7, 11, 0], -1, -84, 0],
[[5, 7, 12, 4], 1, 84, 0],
[[5, 8, 0, 8], 1, 84, 0],
[[5, 8, 1, 7], 1, 84, 0],
[[5, 8, 2, 6], 0, 0, 0],
[[5, 8, 3, 5], -1, -84, 0],
[[5, 8, 4, 4], 1, 84, 0],
[[5, 8, 5, 3], -1, -84, 0],
[[5, 8, 6, 2], -1, -84, 0],
[[5, 8, 7, 1], -1, -84, 0],
[[5, 8, 8, 0], -1, -84, 0],
[[5, 8, 9, 12], 1, 84, 0],
[[5, 8, 10, 11], -1, -84, 0],
[[5, 8, 11, 10], 0, 0, 0],
```

```
[[5, 8, 12, 9], 1, 84, 0],
[[5, 9, 0, 8], 1, 84, 0],
[[5, 9, 1, 2], -1, -84, 0],
[[5, 9, 2, 9], 1, 84, 0],
[[5, 9, 3, 3], -1, -84, 0],
[[5, 9, 4, 10], 0, 0, 0],
[[5, 9, 5, 4], 1, 84, 0],
[[5, 9, 6, 11], -1, -84, 0],
[[5, 9, 7, 5], -1, -84, 0],
[[5, 9, 8, 12], 1, 84, 0],
[[5, 9, 9, 6], 0, 0, 0],
[[5, 9, 10, 0], -1, -84, 0],
[[5, 9, 11, 7], 1, 84, 0],
[[5, 9, 12, 1], -1, -84, 0],
[[5, 10, 0, 8], 1, 84, 0],
[[5, 10, 1, 10], 0, 0, 0],
[[5, 10, 2, 12], 1, 84, 0],
[[5, 10, 3, 1], -1, -84, 0],
[[5, 10, 4, 3], -1, -84, 0],
[[5, 10, 5, 5], -1, -84, 0],
[[5, 10, 6, 7], 1, 84, 0],
[[5, 10, 7, 9], 1, 84, 0],
[[5, 10, 8, 11], -1, -84, 0],
[[5, 10, 9, 0], -1, -84, 0],
[[5, 10, 10, 2], -1, -84, 0],
[[5, 10, 11, 4], 1, 84, 0],
[[5, 10, 12, 6], 0, 0, 0],
[[5, 11, 0, 8], 1, 84, 0],
[[5, 11, 1, 5], -1, -84, 0],
[[5, 11, 2, 2], -1, -84, 0],
[[5, 11, 3, 12], 1, 84, 0],
[[5, 11, 4, 9], 1, 84, 0],
[[5, 11, 5, 6], 0, 0, 0],
[[5, 11, 6, 3], -1, -84, 0],
[[5, 11, 7, 0], -1, -84, 0],
[[5, 11, 8, 10], 0, 0, 0],
[[5, 11, 9, 7], 1, 84, 0],
[[5, 11, 10, 4], 1, 84, 0],
[[5, 11, 11, 1], -1, -84, 0],
[[5, 11, 12, 11], -1, -84, 0],
[[5, 12, 0, 8], 1, 84, 0],
[[5, 12, 1, 0], -1, -84, 0],
[[5, 12, 2, 5], -1, -84, 0],
[[5, 12, 3, 10], 0, 0, 0],
[[5, 12, 4, 2], -1, -84, 0],
[[5, 12, 5, 7], 1, 84, 0],
[[5, 12, 6, 12], 1, 84, 0],
[[5, 12, 7, 4], 1, 84, 0],
[[5, 12, 8, 9], 1, 84, 0],
[[5, 12, 9, 1], -1, -84, 0],
[[5, 12, 10, 6], 0, 0, 0],
[[5, 12, 11, 11], -1, -84, 0],
[[5, 12, 12, 3], -1, -84, 0],
[[6, 0, 0, 11], 1, 84, 0],
[[6, 0, 1, 11], 1, 84, 0],
[[6, 0, 2, 11], 1, 84, 0],
[[6, 0, 3, 11], 1, 84, 0],
[[6, 0, 4, 11], 1, 84, 0],
[[6, 0, 5, 11], 1, 84, 0],
```

```
[[6, 0, 6, 11], 1, 84, 0],
[[6, 0, 7, 11], 1, 84, 0],
[[6, 0, 8, 11], 1, 84, 0],
[[6, 0, 9, 11], 1, 84, 0],
[[6, 0, 10, 11], 1, 84, 0],
[[6, 0, 11, 11], 1, 84, 0],
[[6, 0, 12, 11], 1, 84, 0],
[[6, 1, 0, 11], 1, 84, 0],
[[6, 1, 1, 9], 0, 0, 0],
[[6, 1, 2, 7], 1, 84, 0],
[[6, 1, 3, 5], 0, 0, 0],
[[6, 1, 4, 3], 1, 84, 0],
[[6, 1, 5, 1], -1, -84, 0],
[[6, 1, 6, 12], -1, -84, 0],
[[6, 1, 7, 10], -1, -84, 0],
[[6, 1, 8, 8], 1, 84, 0],
[[6, 1, 9, 6], 1, 84, 0],
[[6, 1, 10, 4], -1, -84, 0],
[[6, 1, 11, 2], -1, -84, 0],
[[6, 1, 12, 0], -1, -84, 0],
[[6, 2, 0, 11], 1, 84, 0],
[[6, 2, 1, 7], 1, 84, 0],
[[6, 2, 2, 3], 1, 84, 0],
[[6, 2, 3, 12], -1, -84, 0],
[[6, 2, 4, 8], 1, 84, 0],
[[6, 2, 5, 4], -1, -84, 0],
[[6, 2, 6, 0], -1, -84, 0],
[[6, 2, 7, 9], 0, 0, 0],
[[6, 2, 8, 5], 0, 0, 0],
[[6, 2, 9, 1], -1, -84, 0],
[[6, 2, 10, 10], -1, -84, 0],
[[6, 2, 11, 6], 1, 84, 0],
[[6, 2, 12, 2], -1, -84, 0],
[[6, 3, 0, 11], 1, 84, 0],
[[6, 3, 1, 5], 0, 0, 0],
[[6, 3, 2, 12], -1, -84, 0],
[[6, 3, 3, 6], 1, 84, 0],
[[6, 3, 4, 0], -1, -84, 0],
[[6, 3, 5, 7], 1, 84, 0],
[[6, 3, 6, 1], -1, -84, 0],
[[6, 3, 7, 8], 1, 84, 0],
[[6, 3, 8, 2], -1, -84, 0],
[[6, 3, 9, 9], 0, 0, 0],
[[6, 3, 10, 3], 1, 84, 0],
[[6, 3, 11, 10], -1, -84, 0],
[[6, 3, 12, 4], -1, -84, 0],
[[6, 4, 0, 11], 1, 84, 0],
[[6, 4, 1, 3], 1, 84, 0],
[[6, 4, 2, 8], 1, 84, 0],
[[6, 4, 3, 0], -1, -84, 0],
[[6, 4, 4, 5], 0, 0, 0],
[[6, 4, 5, 10], -1, -84, 0],
[[6, 4, 6, 2], -1, -84, 0],
[[6, 4, 7, 7], 1, 84, 0],
[[6, 4, 8, 12], -1, -84, 0],
[[6, 4, 9, 4], -1, -84, 0],
[[6, 4, 10, 9], 0, 0, 0],
[[6, 4, 11, 1], -1, -84, 0],
[[6, 4, 12, 6], 1, 84, 0],
```

```
[[6, 5, 0, 11], 1, 84, 0],
[[6, 5, 1, 1], -1, -84, 0],
[[6, 5, 2, 4], -1, -84, 0],
[[6, 5, 3, 7], 1, 84, 0],
[[6, 5, 4, 10], -1, -84, 0],
[[6, 5, 5, 0], -1, -84, 0],
[[6, 5, 6, 3], 1, 84, 0],
[[6, 5, 7, 6], 1, 84, 0],
[[6, 5, 8, 9], 0, 0, 0],
[[6, 5, 9, 12], -1, -84, 0],
[[6, 5, 10, 2], -1, -84, 0],
[[6, 5, 11, 5], 0, 0, 0],
[[6, 5, 12, 8], 1, 84, 0],
[[6, 6, 0, 11], 1, 84, 0],
[[6, 6, 1, 12], -1, -84, 0],
[[6, 6, 2, 0], -1, -84, 0],
[[6, 6, 3, 1], -1, -84, 0],
[[6, 6, 4, 2], -1, -84, 0],
[[6, 6, 5, 3], 1, 84, 0],
[[6, 6, 6, 4], -1, -84, 0],
[[6, 6, 7, 5], 0, 0, 0],
[[6, 6, 8, 6], 1, 84, 0],
[[6, 6, 9, 7], 1, 84, 0],
[[6, 6, 10, 8], 1, 84, 0],
[[6, 6, 11, 9], 0, 0, 0],
[[6, 6, 12, 10], -1, -84, 0],
[[6, 7, 0, 11], 1, 84, 0],
[[6, 7, 1, 10], -1, -84, 0],
[[6, 7, 2, 9], 0, 0, 0],
[[6, 7, 3, 8], 1, 84, 0],
[[6, 7, 4, 7], 1, 84, 0],
[[6, 7, 5, 6], 1, 84, 0],
[[6, 7, 6, 5], 0, 0, 0],
[[6, 7, 7, 4], -1, -84, 0],
[[6, 7, 8, 3], 1, 84, 0],
[[6, 7, 9, 2], -1, -84, 0],
[[6, 7, 10, 1], -1, -84, 0],
[[6, 7, 11, 0], -1, -84, 0],
[[6, 7, 12, 12], -1, -84, 0],
[[6, 8, 0, 11], 1, 84, 0],
[[6, 8, 1, 8], 1, 84, 0],
[[6, 8, 2, 5], 0, 0, 0],
[[6, 8, 3, 2], -1, -84, 0],
[[6, 8, 4, 12], -1, -84, 0],
[[6, 8, 5, 9], 0, 0, 0],
[[6, 8, 6, 6], 1, 84, 0],
[[6, 8, 7, 3], 1, 84, 0],
[[6, 8, 8, 0], -1, -84, 0],
[[6, 8, 9, 10], -1, -84, 0],
[[6, 8, 10, 7], 1, 84, 0],
[[6, 8, 11, 4], -1, -84, 0],
[[6, 8, 12, 1], -1, -84, 0],
[[6, 9, 0, 11], 1, 84, 0],
[[6, 9, 1, 6], 1, 84, 0],
[[6, 9, 2, 1], -1, -84, 0],
[[6, 9, 3, 9], 0, 0, 0],
[[6, 9, 4, 4], -1, -84, 0],
[[6, 9, 5, 12], -1, -84, 0],
[[6, 9, 6, 7], 1, 84, 0],
```

```
        [[6, 9, 7, 2], -1, -84, 0],
        [[6, 9, 8, 10], -1, -84, 0],
        [[6, 9, 9, 5], 0, 0, 0],
        [[6, 9, 10, 0], -1, -84, 0],
        [[6, 9, 11, 8], 1, 84, 0],
        [[6, 9, 12, 3], 1, 84, 0],
        [[6, 10, 0, 11], 1, 84, 0],
        [[6, 10, 1, 4], -1, -84, 0],
        [[6, 10, 2, 10], -1, -84, 0],
        [[6, 10, 3, 3], 1, 84, 0],
        [[6, 10, 4, 9], 0, 0, 0],
        [[6, 10, 5, 2], -1, -84, 0],
        [[6, 10, 6, 8], 1, 84, 0],
        [[6, 10, 7, 1], -1, -84, 0],
        [[6, 10, 8, 7], 1, 84, 0],
        [[6, 10, 9, 0], -1, -84, 0],
        [[6, 10, 10, 6], 1, 84, 0],
        [[6, 10, 11, 12], -1, -84, 0],
        [[6, 10, 12, 5], 0, 0, 0],
        [[6, 11, 0, 11], 1, 84, 0],
        [[6, 11, 1, 2], -1, -84, 0],
        [[6, 11, 2, 6], 1, 84, 0],
        [[6, 11, 3, 10], -1, -84, 0],
        [[6, 11, 4, 1], -1, -84, 0],
        [[6, 11, 5, 5], 0, 0, 0],
        [[6, 11, 6, 9], 0, 0, 0],
        [[6, 11, 7, 0], -1, -84, 0],
        [[6, 11, 8, 4], -1, -84, 0],
        [[6, 11, 9, 8], 1, 84, 0],
        [[6, 11, 10, 12], -1, -84, 0],
        [[6, 11, 11, 3], 1, 84, 0],
        [[6, 11, 12, 7], 1, 84, 0],
        [[6, 12, 0, 11], 1, 84, 0],
        [[6, 12, 1, 0], -1, -84, 0],
        [[6, 12, 2, 2], -1, -84, 0],
        [[6, 12, 3, 4], -1, -84, 0],
        [[6, 12, 4, 6], 1, 84, 0],
        [[6, 12, 5, 8], 1, 84, 0],
        [[6, 12, 6, 10], -1, -84, 0],
        [[6, 12, 7, 12], -1, -84, 0],
        [[6, 12, 8, 1], -1, -84, 0],
        [[6, 12, 9, 3], 1, 84, 0],
        [[6, 12, 10, 5], 0, 0, 0],
        [[6, 12, 11, 7], 1, 84, 0],
        [[6, 12, 12, 9], 0, 0, 0]
      ]
    }
  ],
  "characters": [
    {
      "p": 5,
      "tail_solutions": [[4, 0]],
      "igusa_solutions": [[6, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": []
    },
    {
      "p": 7,
```

```
      "tail_solutions": [[6, 0]],
      "igusa_solutions": [[8, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [2, 3, 4, 5]
    },
    {
      "p": 11,
      "tail_solutions": [[10, 0]],
      "igusa_solutions": [[12, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [2, 3, 4, 5, 6, 7, 8, 9]
    },
    {
      "p": 13,
      "tail_solutions": [[12, 0]],
      "igusa_solutions": [[14, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [2, 3, 4, 6, 7, 9, 10, 11]
    },
    {
      "p": 17,
      "tail_solutions": [[16, 0]],
      "igusa_solutions": [[18, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15]
    },
    {
      "p": 19,
      "tail_solutions": [[18, 0]],
      "igusa_solutions": [[20, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17]
    },
    {
      "p": 23,
      "tail_solutions": [[22, 0]],
      "igusa_solutions": [[24, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21]
    },
    {
      "p": 29,
      "tail_solutions": [[28, 0]],
      "igusa_solutions": [[30, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [
        2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24,
        25, 26, 27
      ]
    },
```

```
{
  "p": 31,
  "tail_solutions": [[30, 0]],
  "igusa_solutions": [[32, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 27, 28, 29
  ]
},
{
  "p": 37,
  "tail_solutions": [[36, 0]],
  "igusa_solutions": [[38, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
    24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35
  ]
},
{
  "p": 41,
  "tail_solutions": [[40, 0]],
  "igusa_solutions": [[42, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
    24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39
  ]
},
{
  "p": 43,
  "tail_solutions": [[42, 0]],
  "igusa_solutions": [[44, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41
  ]
},
{
  "p": 47,
  "tail_solutions": [[46, 0]],
  "igusa_solutions": [[48, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
    42, 43, 44, 45
  ]
},
{
  "p": 53,
  "tail_solutions": [[52, 0]],
```

```
    "igusa_solutions": [[54, 2]],
    "induced_invariant_dimension": 0,
    "correct_scalar_bad_residues": [],
    "old_scalar_nonscalar_residues": [
      2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
      24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
      44, 45, 46, 47, 48, 49, 50, 51
    ]
  },
  {
    "p": 59,
    "tail_solutions": [[58, 0]],
    "igusa_solutions": [[60, 2]],
    "induced_invariant_dimension": 0,
    "correct_scalar_bad_residues": [],
    "old_scalar_nonscalar_residues": [
      2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
      23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
      42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57
    ]
  },
  {
    "p": 61,
    "tail_solutions": [[60, 0]],
    "igusa_solutions": [[62, 2]],
    "induced_invariant_dimension": 0,
    "correct_scalar_bad_residues": [],
    "old_scalar_nonscalar_residues": [
      2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
      24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42,
      43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 59
    ]
  },
  {
    "p": 67,
    "tail_solutions": [[66, 0]],
    "igusa_solutions": [[68, 2]],
    "induced_invariant_dimension": 0,
    "correct_scalar_bad_residues": [],
    "old_scalar_nonscalar_residues": [
      2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
      23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
      42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
      61, 62, 63, 64, 65
    ]
  },
  {
    "p": 71,
    "tail_solutions": [[70, 0]],
    "igusa_solutions": [[72, 2]],
    "induced_invariant_dimension": 0,
    "correct_scalar_bad_residues": [],
    "old_scalar_nonscalar_residues": [
      2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
      23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
      42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
      61, 62, 63, 64, 65, 66, 67, 68, 69
    ]
  },
```

```
{
  "p": 73,
  "tail_solutions": [[72, 0]],
  "igusa_solutions": [[74, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42,
    43, 44, 45, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
    63, 64, 65, 66, 67, 68, 69, 70, 71
  ]
},
{
  "p": 79,
  "tail_solutions": [[78, 0]],
  "igusa_solutions": [[80, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
    42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
    61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77
  ]
},
{
  "p": 83,
  "tail_solutions": [[82, 0]],
  "igusa_solutions": [[84, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
    42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
    61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
    80, 81
  ]
},
{
  "p": 89,
  "tail_solutions": [[88, 0]],
  "igusa_solutions": [[90, 2]],
  "induced_invariant_dimension": 0,
  "correct_scalar_bad_residues": [],
  "old_scalar_nonscalar_residues": [
    2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
    23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42,
    43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 56, 57, 58, 59, 60, 61, 62,
    63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
    82, 83, 84, 85, 86, 87
  ]
},
{
  "p": 97,
  "tail_solutions": [[96, 0]],
  "igusa_solutions": [[98, 2]],
  "induced_invariant_dimension": 0,
```

```
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [
        2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23,
        24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42,
        43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
        62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79, 80, 81,
        82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95
      ]
    },
    {
      "p": 101,
      "tail_solutions": [[100, 0]],
      "igusa_solutions": [[102, 2]],
      "induced_invariant_dimension": 0,
      "correct_scalar_bad_residues": [],
      "old_scalar_nonscalar_residues": [
        2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
        24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42,
        43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
        62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
        81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 92, 93, 94, 95, 96, 97, 98, 99
      ]
    }
  ],
  "cusp_order_exact_rank": [
    {
      "p": 5,
      "n": 2,
      "M": 1,
      "rows": [
        [0, 1, null, 1, 1],
        [1, 1, 1, 1, 5],
        [1, 1, 2, 2, 5],
        [1, 1, 3, 3, 5],
        [1, 1, 4, 4, 5],
        [2, 1, null, 1, 25]
      ],
      "columns": [
        ["ordinary", 5, null],
        ["ordinary", 25, null],
        ["shift", 1, 1],
        ["shift", 1, 2],
        ["shift", 1, 3]
      ],
      "matrix": [
        [-20, -24, 0, 0, 0],
        [4, 0, -24, -24, -24],
        [4, 0, 0, 0, 24],
        [4, 0, 0, 24, 0],
        [4, 0, 24, 0, 0],
        [4, 24, 0, 0, 0]
      ],
      "deleted_row": 5,
      "square_minor_determinant": {"sign": -1, "decimal_blocks_to_concatenate":
["5308416"]},
      "rank": 5
    },
    {
      "p": 5,
```

```
"n": 2,
"M": 4,
"rows": [
  [0, 1, null, 1, 1],
  [0, 2, null, 1, 2],
  [0, 4, null, 1, 4],
  [1, 1, 1, 1, 5],
  [1, 1, 2, 2, 5],
  [1, 1, 3, 3, 5],
  [1, 1, 4, 4, 5],
  [1, 2, 1, 7, 10],
  [1, 2, 2, 9, 10],
  [1, 2, 3, 1, 10],
  [1, 2, 4, 3, 10],
  [1, 4, 1, 9, 20],
  [1, 4, 2, 3, 20],
  [1, 4, 3, 7, 20],
  [1, 4, 4, 1, 20],
  [2, 1, null, 1, 25],
  [2, 2, null, 1, 50],
  [2, 4, null, 1, 100]
],
"columns": [
  ["ordinary", 2, null],
  ["ordinary", 4, null],
  ["ordinary", 5, null],
  ["ordinary", 10, null],
  ["ordinary", 20, null],
  ["ordinary", 25, null],
  ["ordinary", 50, null],
  ["ordinary", 100, null],
  ["shift", 1, 1],
  ["shift", 1, 2],
  ["shift", 1, 3],
  ["shift", 2, 1],
  ["shift", 2, 2],
  ["shift", 2, 3],
  ["shift", 4, 1],
  ["shift", 4, 2],
  ["shift", 4, 3]
],
"matrix": [
  [-50, -75, -80, -90, -95, -96, -98, -99, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [25, 0, -20, -15, -20, -24, -23, -24, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [25, 75, -20, -15, -5, -24, -23, -21, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [-2, -3, 16, 6, 1, 0, -2, -3, -96, -96, -96, 0, 0, 48, 24, 0, 0],
  [-2, -3, 16, 6, 1, 0, -2, -3, 0, 0, 96, 48, 0, 0, 0, 24, 0],
  [-2, -3, 16, 6, 1, 0, -2, -3, 0, 96, 0, -48, -48, -48, 0, 0, 24],
  [-2, -3, 16, 6, 1, 0, -2, -3, 96, 0, 0, 0, 48, 0, -24, -24, -24],
  [1, 0, 4, 9, 4, 0, 1, 0, -24, -24, -24, 0, 0, 48, 24, 0, 0],
  [1, 0, 4, 9, 4, 0, 1, 0, 0, 0, 24, 48, 0, 0, 0, 24, 0],
  [1, 0, 4, 9, 4, 0, 1, 0, 0, 24, 0, -48, -48, -48, 0, 0, 24],
  [1, 0, 4, 9, 4, 0, 1, 0, 24, 0, 0, 0, 48, 0, -24, -24, -24],
  [1, 3, 4, 9, 19, 0, 1, 3, -24, -24, -24, 0, 0, 48, 96, 0, 0],
  [1, 3, 4, 9, 19, 0, 1, 3, 0, 0, 24, 48, 0, 0, 0, 96, 0],
  [1, 3, 4, 9, 19, 0, 1, 3, 0, 24, 0, -48, -48, -48, 0, 0, 96],
  [1, 3, 4, 9, 19, 0, 1, 3, 24, 0, 0, 0, 48, 0, -96, -96, -96],
  [-2, -3, 16, 6, 1, 96, 46, 21, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [1, 0, 4, 9, 4, 24, 49, 24, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
      [1, 3, 4, 9, 19, 24, 49, 99, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    ],
    "deleted_row": 17,
    "square_minor_determinant": {
      "sign": -1,
      "decimal_blocks_to_concatenate": ["763169733152552103681038745600"]
    },
    "rank": 17
  },
  {
    "p": 5,
    "n": 3,
    "M": 6,
    "rows": [
      [0, 1, null, 1, 1],
      [0, 2, null, 1, 2],
      [0, 3, null, 1, 3],
      [0, 6, null, 1, 6],
      [1, 1, 1, 1, 5],
      [1, 1, 2, 2, 5],
      [1, 1, 3, 3, 5],
      [1, 1, 4, 4, 5],
      [1, 2, 1, 7, 10],
      [1, 2, 2, 9, 10],
      [1, 2, 3, 1, 10],
      [1, 2, 4, 3, 10],
      [1, 3, 1, 8, 15],
      [1, 3, 2, 1, 15],
      [1, 3, 3, 4, 15],
      [1, 3, 4, 2, 15],
      [1, 6, 1, 1, 30],
      [1, 6, 2, 7, 30],
      [1, 6, 3, 13, 30],
      [1, 6, 4, 19, 30],
      [2, 1, 1, 1, 25],
      [2, 1, 2, 2, 25],
      [2, 1, 3, 3, 25],
      [2, 1, 4, 4, 25],
      [2, 2, 1, 7, 50],
      [2, 2, 2, 9, 50],
      [2, 2, 3, 1, 50],
      [2, 2, 4, 3, 50],
      [2, 3, 1, 8, 75],
      [2, 3, 2, 1, 75],
      [2, 3, 3, 4, 75],
      [2, 3, 4, 2, 75],
      [2, 6, 1, 1, 150],
      [2, 6, 2, 7, 150],
      [2, 6, 3, 13, 150],
      [2, 6, 4, 19, 150],
      [3, 1, null, 1, 125],
      [3, 2, null, 1, 250],
      [3, 3, null, 1, 375],
      [3, 6, null, 1, 750]
    ],
    "columns": [
      ["ordinary", 2, null],
      ["ordinary", 3, null],
      ["ordinary", 5, null],
```

```
      ["ordinary", 6, null],
      ["ordinary", 10, null],
      ["ordinary", 15, null],
      ["ordinary", 25, null],
      ["ordinary", 30, null],
      ["ordinary", 50, null],
      ["ordinary", 75, null],
      ["ordinary", 125, null],
      ["ordinary", 150, null],
      ["ordinary", 250, null],
      ["ordinary", 375, null],
      ["ordinary", 750, null],
      ["shift", 1, 1],
      ["shift", 1, 2],
      ["shift", 1, 3],
      ["shift", 2, 1],
      ["shift", 2, 2],
      ["shift", 2, 3],
      ["shift", 3, 1],
      ["shift", 3, 2],
      ["shift", 3, 3],
      ["shift", 6, 1],
      ["shift", 6, 2],
      ["shift", 6, 3],
      ["shift", 5, 1],
      ["shift", 5, 2],
      ["shift", 5, 3],
      ["shift", 10, 1],
      ["shift", 10, 2],
      ["shift", 10, 3],
      ["shift", 15, 1],
      ["shift", 15, 2],
      ["shift", 15, 3],
      ["shift", 30, 1],
      ["shift", 30, 2],
      ["shift", 30, 3]
    ],
    "matrix": [
      [
        -375, -500, -600, -625, -675, -700, -720, -725, -735, -740, -744, -745,
        -747, -748, -749, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0
      ],
      [
        375, -250, -300, -125, -225, -350, -360, -325, -345, -370, -372, -365, -369,
        -374, -373, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0
      ],
      [
        -125, 500, -200, 125, -225, -100, -240, -175, -245, -220, -248, -235, -249,
        -244, -247, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0
      ],
      [
        125, 250, -100, 625, -75, -50, -120, 25, -115, -110, -124, -95, -123, -122,
        -119, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ],
      [
        -15, -20, 120, -25, 45, 20, 0, -5, -15, -20, -24, -25, -27, -28, -29, -720,
```

```
  -720, -720, 0, 0, 360, 0, 240, 0, -120, -120, -120, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0
],
[
  -15, -20, 120, -25, 45, 20, 0, -5, -15, -20, -24, -25, -27, -28, -29, 0, 0,
  720, 360, 0, 0, -240, -240, -240, 0, 0, 120, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0
],
[
  -15, -20, 120, -25, 45, 20, 0, -5, -15, -20, -24, -25, -27, -28, -29, 0,
  720, 0, -360, -360, -360, 240, 0, 0, 0, 120, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0
],
[
  -15, -20, 120, -25, 45, 20, 0, -5, -15, -20, -24, -25, -27, -28, -29, 720,
  0, 0, 0, 360, 0, 0, 0, 240, 120, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  15, -10, 60, -5, 135, 10, 0, 35, 15, -10, -12, -5, -9, -14, -13, -360, -360,
  -360, 0, 0, 720, 0, 120, 0, -240, -240, -240, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0
],
[
  15, -10, 60, -5, 135, 10, 0, 35, 15, -10, -12, -5, -9, -14, -13, 0, 0, 360,
  720, 0, 0, -120, -120, -120, 0, 0, 240, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  15, -10, 60, -5, 135, 10, 0, 35, 15, -10, -12, -5, -9, -14, -13, 0, 360, 0,
  -720, -720, -720, 120, 0, 0, 0, 240, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  15, -10, 60, -5, 135, 10, 0, 35, 15, -10, -12, -5, -9, -14, -13, 360, 0, 0,
  0, 720, 0, 0, 0, 120, 240, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  -5, 20, 40, 5, 15, 140, 0, 65, -5, 20, -8, 5, -9, -4, -7, -240, -240, -240,
  0, 0, 120, 0, 720, 0, -360, -360, -360, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  -5, 20, 40, 5, 15, 140, 0, 65, -5, 20, -8, 5, -9, -4, -7, 0, 0, 240, 120, 0,
  0, -720, -720, -720, 0, 0, 360, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  -5, 20, 40, 5, 15, 140, 0, 65, -5, 20, -8, 5, -9, -4, -7, 0, 240, 0, -120,
  -120, -120, 720, 0, 0, 0, 360, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  -5, 20, 40, 5, 15, 140, 0, 65, -5, 20, -8, 5, -9, -4, -7, 240, 0, 0, 0, 120,
  0, 0, 0, 720, 360, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  5, 10, 20, 25, 45, 70, 0, 145, 5, 10, -4, 25, -3, -2, 1, -120, -120, -120,
  0, 0, 240, 0, 360, 0, -720, -720, -720, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  5, 10, 20, 25, 45, 70, 0, 145, 5, 10, -4, 25, -3, -2, 1, 0, 0, 120, 240, 0,
  0, -360, -360, -360, 0, 0, 720, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
```

```
  5, 10, 20, 25, 45, 70, 0, 145, 5, 10, -4, 25, -3, -2, 1, 0, 120, 0, -240,
  -240, -240, 360, 0, 0, 0, 720, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  5, 10, 20, 25, 45, 70, 0, 145, 5, 10, -4, 25, -3, -2, 1, 120, 0, 0, 0, 240,
  0, 0, 0, 360, 720, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  -3, -4, 24, -5, 9, 4, 144, -1, 69, 44, 24, 19, 9, 4, -1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, -720, -720, -720, 0, 0, 360, 0, 240, 0, -120, -120, -120
],
[
  -3, -4, 24, -5, 9, 4, 144, -1, 69, 44, 24, 19, 9, 4, -1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 720, 360, 0, 0, -240, -240, -240, 0, 0, 120
],
[
  -3, -4, 24, -5, 9, 4, 144, -1, 69, 44, 24, 19, 9, 4, -1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 720, 0, -360, -360, -360, 240, 0, 0, 0, 120, 0
],
[
  -3, -4, 24, -5, 9, 4, 144, -1, 69, 44, 24, 19, 9, 4, -1, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 720, 0, 0, 0, 360, 0, 0, 0, 240, 120, 0, 0
],
[
  3, -2, 12, -1, 27, 2, 72, 7, 147, 22, 12, 47, 27, 2, 7, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, -360, -360, -360, 0, 0, 720, 0, 120, 0, -240, -240, -240
],
[
  3, -2, 12, -1, 27, 2, 72, 7, 147, 22, 12, 47, 27, 2, 7, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 360, 720, 0, 0, -120, -120, -120, 0, 0, 240
],
[
  3, -2, 12, -1, 27, 2, 72, 7, 147, 22, 12, 47, 27, 2, 7, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 360, 0, -720, -720, -720, 120, 0, 0, 0, 240, 0
],
[
  3, -2, 12, -1, 27, 2, 72, 7, 147, 22, 12, 47, 27, 2, 7, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 360, 0, 0, 0, 720, 0, 0, 0, 120, 240, 0, 0
],
[
  -1, 4, 8, 1, 3, 28, 48, 13, 23, 148, 8, 73, 3, 28, 13, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, -240, -240, -240, 0, 0, 120, 0, 720, 0, -360, -360, -360
],
[
  -1, 4, 8, 1, 3, 28, 48, 13, 23, 148, 8, 73, 3, 28, 13, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 240, 120, 0, 0, -720, -720, -720, 0, 0, 360
],
[
  -1, 4, 8, 1, 3, 28, 48, 13, 23, 148, 8, 73, 3, 28, 13, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 240, 0, -120, -120, -120, 720, 0, 0, 0, 360, 0
],
[
  -1, 4, 8, 1, 3, 28, 48, 13, 23, 148, 8, 73, 3, 28, 13, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 240, 0, 0, 0, 120, 0, 0, 0, 720, 360, 0, 0
],
[
  1, 2, 4, 5, 9, 14, 24, 29, 49, 74, 4, 149, 9, 14, 29, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, -120, -120, -120, 0, 0, 240, 0, 360, 0, -720, -720, -720
],
```

```
      [
        1, 2, 4, 5, 9, 14, 24, 29, 49, 74, 4, 149, 9, 14, 29, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 120, 240, 0, 0, -360, -360, -360, 0, 0, 720
      ],
      [
        1, 2, 4, 5, 9, 14, 24, 29, 49, 74, 4, 149, 9, 14, 29, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 120, 0, -240, -240, -240, 360, 0, 0, 0, 720, 0
      ],
      [
        1, 2, 4, 5, 9, 14, 24, 29, 49, 74, 4, 149, 9, 14, 29, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 120, 0, 0, 0, 240, 0, 0, 0, 360, 720, 0, 0
      ],
      [
        -3, -4, 24, -5, 9, 4, 144, -1, 69, 44, 744, 19, 369, 244, 119, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ],
      [
        3, -2, 12, -1, 27, 2, 72, 7, 147, 22, 372, 47, 747, 122, 247, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ],
      [
        -1, 4, 8, 1, 3, 28, 48, 13, 23, 148, 248, 73, 123, 748, 373, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ],
      [
        1, 2, 4, 5, 9, 14, 24, 29, 49, 74, 124, 149, 249, 374, 749, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ]
    ],
    "deleted_row": 39,
    "square_minor_determinant": {
      "sign": 1,
      "decimal_blocks_to_concatenate": [
        "64839123413318205933506542387856295578503484028609584758572 8",
        "95646515200000000000000000000000000000000000000000"
      ]
    },
    "rank": 39
  },
  {
    "p": 7,
    "n": 2,
    "M": 4,
    "rows": [
      [0, 1, null, 1, 1],
      [0, 2, null, 1, 2],
      [0, 4, null, 1, 4],
      [1, 1, 1, 1, 7],
      [1, 1, 2, 2, 7],
      [1, 1, 3, 3, 7],
      [1, 1, 4, 4, 7],
      [1, 1, 5, 5, 7],
      [1, 1, 6, 6, 7],
      [1, 2, 1, 9, 14],
      [1, 2, 2, 11, 14],
      [1, 2, 3, 13, 14],
      [1, 2, 4, 1, 14],
      [1, 2, 5, 3, 14],
      [1, 2, 6, 5, 14],
```

```
    [1, 4, 1, 11, 28],
    [1, 4, 2, 1, 28],
    [1, 4, 3, 5, 28],
    [1, 4, 4, 9, 28],
    [1, 4, 5, 13, 28],
    [1, 4, 6, 3, 28],
    [2, 1, null, 1, 49],
    [2, 2, null, 1, 98],
    [2, 4, null, 1, 196]
  ],
  "columns": [
    ["ordinary", 2, null],
    ["ordinary", 4, null],
    ["ordinary", 7, null],
    ["ordinary", 14, null],
    ["ordinary", 28, null],
    ["ordinary", 49, null],
    ["ordinary", 98, null],
    ["ordinary", 196, null],
    ["shift", 1, 1],
    ["shift", 1, 2],
    ["shift", 1, 3],
    ["shift", 1, 4],
    ["shift", 1, 5],
    ["shift", 2, 1],
    ["shift", 2, 2],
    ["shift", 2, 3],
    ["shift", 2, 4],
    ["shift", 2, 5],
    ["shift", 4, 1],
    ["shift", 4, 2],
    ["shift", 4, 3],
    ["shift", 4, 4],
    ["shift", 4, 5]
  ],
  "matrix": [
    [
      -98, -147, -168, -182, -189, -192, -194, -195, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0
    ],
    [
      49, 0, -42, -35, -42, -48, -47, -48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0
    ],
    [
      49, 147, -42, -35, -21, -48, -47, -45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0
    ],
    [
      -2, -3, 24, 10, 3, 0, -2, -3, -192, -192, -192, -192, -192, 0, 0, 0, 0, 96,
      0, 0, 48, 0, 0
    ],
    [
      -2, -3, 24, 10, 3, 0, -2, -3, 0, 0, 0, 0, 192, 0, 0, 96, 0, 0, -48, -48,
      -48, -48, -48
    ],
    [
      -2, -3, 24, 10, 3, 0, -2, -3, 0, 0, 0, 192, 0, 96, 0, 0, 0, 0, 0, 48, 0, 0,
      0
```

```
        ],
        [
          -2, -3, 24, 10, 3, 0, -2, -3, 0, 0, 192, 0, 0, -96, -96, -96, -96, -96, 0,
          0, 0, 0, 48
        ],
        [
          -2, -3, 24, 10, 3, 0, -2, -3, 0, 192, 0, 0, 0, 0, 0, 0, 96, 0, 48, 0, 0, 0,
          0
        ],
        [
          -2, -3, 24, 10, 3, 0, -2, -3, 192, 0, 0, 0, 0, 0, 0, 96, 0, 0, 0, 0, 0, 0, 48,
          0
        ],
        [
          1, 0, 6, 13, 6, 0, 1, 0, -48, -48, -48, -48, -48, 0, 0, 0, 0, 96, 0, 0, 48,
          0, 0
        ],
        [
          1, 0, 6, 13, 6, 0, 1, 0, 0, 0, 0, 0, 48, 0, 0, 96, 0, 0, -48, -48, -48, -48,
          -48
        ],
        [1, 0, 6, 13, 6, 0, 1, 0, 0, 0, 0, 48, 0, 96, 0, 0, 0, 0, 0, 48, 0, 0, 0],
        [
          1, 0, 6, 13, 6, 0, 1, 0, 0, 0, 48, 0, 0, -96, -96, -96, -96, -96, 0, 0, 0,
          0, 48
        ],
        [1, 0, 6, 13, 6, 0, 1, 0, 0, 48, 0, 0, 0, 0, 0, 0, 96, 0, 48, 0, 0, 0, 0],
        [1, 0, 6, 13, 6, 0, 1, 0, 48, 0, 0, 0, 0, 0, 96, 0, 0, 0, 0, 0, 0, 48, 0],
        [
          1, 3, 6, 13, 27, 0, 1, 3, -48, -48, -48, -48, -48, 0, 0, 0, 0, 96, 0, 0,
          192, 0, 0
        ],
        [
          1, 3, 6, 13, 27, 0, 1, 3, 0, 0, 0, 0, 48, 0, 0, 96, 0, 0, -192, -192, -192,
          -192, -192
        ],
        [1, 3, 6, 13, 27, 0, 1, 3, 0, 0, 0, 48, 0, 96, 0, 0, 0, 0, 0, 192, 0, 0, 0],
        [
          1, 3, 6, 13, 27, 0, 1, 3, 0, 0, 48, 0, 0, -96, -96, -96, -96, -96, 0, 0, 0,
          0, 192
        ],
        [1, 3, 6, 13, 27, 0, 1, 3, 0, 48, 0, 0, 0, 0, 0, 0, 96, 0, 192, 0, 0, 0, 0],
        [1, 3, 6, 13, 27, 0, 1, 3, 48, 0, 0, 0, 0, 0, 0, 96, 0, 0, 0, 0, 0, 0, 192, 0],
        [-2, -3, 24, 10, 3, 192, 94, 45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 0, 6, 13, 6, 48, 97, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 3, 6, 13, 27, 48, 97, 195, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
      ],
      "deleted_row": 23,
      "square_minor_determinant": {
        "sign": -1,
        "decimal_blocks_to_concatenate":
["491648387962779211629602284856231854231397597184"]
      },
      "rank": 23
    },
    {
      "p": 7,
      "n": 3,
      "M": 4,
```

```
"rows": [
  [0, 1, null, 1, 1],
  [0, 2, null, 1, 2],
  [0, 4, null, 1, 4],
  [1, 1, 1, 1, 7],
  [1, 1, 2, 2, 7],
  [1, 1, 3, 3, 7],
  [1, 1, 4, 4, 7],
  [1, 1, 5, 5, 7],
  [1, 1, 6, 6, 7],
  [1, 2, 1, 9, 14],
  [1, 2, 2, 11, 14],
  [1, 2, 3, 13, 14],
  [1, 2, 4, 1, 14],
  [1, 2, 5, 3, 14],
  [1, 2, 6, 5, 14],
  [1, 4, 1, 11, 28],
  [1, 4, 2, 1, 28],
  [1, 4, 3, 5, 28],
  [1, 4, 4, 9, 28],
  [1, 4, 5, 13, 28],
  [1, 4, 6, 3, 28],
  [2, 1, 1, 1, 49],
  [2, 1, 2, 2, 49],
  [2, 1, 3, 3, 49],
  [2, 1, 4, 4, 49],
  [2, 1, 5, 5, 49],
  [2, 1, 6, 6, 49],
  [2, 2, 1, 9, 98],
  [2, 2, 2, 11, 98],
  [2, 2, 3, 13, 98],
  [2, 2, 4, 1, 98],
  [2, 2, 5, 3, 98],
  [2, 2, 6, 5, 98],
  [2, 4, 1, 11, 196],
  [2, 4, 2, 1, 196],
  [2, 4, 3, 5, 196],
  [2, 4, 4, 9, 196],
  [2, 4, 5, 13, 196],
  [2, 4, 6, 3, 196],
  [3, 1, null, 1, 343],
  [3, 2, null, 1, 686],
  [3, 4, null, 1, 1372]
],
"columns": [
  ["ordinary", 2, null],
  ["ordinary", 4, null],
  ["ordinary", 7, null],
  ["ordinary", 14, null],
  ["ordinary", 28, null],
  ["ordinary", 49, null],
  ["ordinary", 98, null],
  ["ordinary", 196, null],
  ["ordinary", 343, null],
  ["ordinary", 686, null],
  ["ordinary", 1372, null],
  ["shift", 1, 1],
  ["shift", 1, 2],
  ["shift", 1, 3],
```

```
    ["shift", 1, 4],
    ["shift", 1, 5],
    ["shift", 2, 1],
    ["shift", 2, 2],
    ["shift", 2, 3],
    ["shift", 2, 4],
    ["shift", 2, 5],
    ["shift", 4, 1],
    ["shift", 4, 2],
    ["shift", 4, 3],
    ["shift", 4, 4],
    ["shift", 4, 5],
    ["shift", 7, 1],
    ["shift", 7, 2],
    ["shift", 7, 3],
    ["shift", 7, 4],
    ["shift", 7, 5],
    ["shift", 14, 1],
    ["shift", 14, 2],
    ["shift", 14, 3],
    ["shift", 14, 4],
    ["shift", 14, 5],
    ["shift", 28, 1],
    ["shift", 28, 2],
    ["shift", 28, 3],
    ["shift", 28, 4],
    ["shift", 28, 5]
  ],
  "matrix": [
    [
      -686, -1029, -1176, -1274, -1323, -1344, -1358, -1365, -1368, -1370, -1371,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0
    ],
    [
      343, 0, -294, -245, -294, -336, -329, -336, -342, -341, -342, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    ],
    [
      343, 1029, -294, -245, -147, -336, -329, -315, -342, -341, -339, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    ],
    [
      -14, -21, 168, 70, 21, 0, -14, -21, -24, -26, -27, -1344, -1344, -1344,
      -1344, -1344, 0, 0, 0, 0, 672, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0
    ],
    [
      -14, -21, 168, 70, 21, 0, -14, -21, -24, -26, -27, 0, 0, 0, 0, 1344, 0, 0,
      672, 0, 0, -336, -336, -336, -336, -336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0
    ],
    [
      -14, -21, 168, 70, 21, 0, -14, -21, -24, -26, -27, 0, 0, 0, 1344, 0, 672, 0,
      0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    ],
    [
      -14, -21, 168, 70, 21, 0, -14, -21, -24, -26, -27, 0, 0, 1344, 0, 0, -672,
      -672, -672, -672, -672, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
  0, 0, 0
],
[
  -14, -21, 168, 70, 21, 0, -14, -21, -24, -26, -27, 0, 1344, 0, 0, 0, 0, 0,
  0, 672, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  -14, -21, 168, 70, 21, 0, -14, -21, -24, -26, -27, 1344, 0, 0, 0, 0, 0, 672,
  0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 0, 42, 91, 42, 0, 7, 0, -6, -5, -6, -336, -336, -336, -336, -336, 0, 0,
  0, 0, 672, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 0, 42, 91, 42, 0, 7, 0, -6, -5, -6, 0, 0, 0, 0, 336, 0, 0, 672, 0, 0,
  -336, -336, -336, -336, -336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 0, 42, 91, 42, 0, 7, 0, -6, -5, -6, 0, 0, 0, 336, 0, 672, 0, 0, 0, 0, 0,
  336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 0, 42, 91, 42, 0, 7, 0, -6, -5, -6, 0, 0, 336, 0, 0, -672, -672, -672,
  -672, -672, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 0, 42, 91, 42, 0, 7, 0, -6, -5, -6, 0, 336, 0, 0, 0, 0, 0, 0, 672, 0,
  336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 0, 42, 91, 42, 0, 7, 0, -6, -5, -6, 336, 0, 0, 0, 0, 0, 672, 0, 0, 0, 0,
  0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 21, 42, 91, 189, 0, 7, 21, -6, -5, -3, -336, -336, -336, -336, -336, 0,
  0, 0, 0, 672, 0, 0, 1344, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 21, 42, 91, 189, 0, 7, 21, -6, -5, -3, 0, 0, 0, 0, 336, 0, 0, 672, 0, 0,
  -1344, -1344, -1344, -1344, -1344, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0
],
[
  7, 21, 42, 91, 189, 0, 7, 21, -6, -5, -3, 0, 0, 0, 336, 0, 672, 0, 0, 0, 0,
  0, 1344, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 21, 42, 91, 189, 0, 7, 21, -6, -5, -3, 0, 0, 336, 0, 0, -672, -672, -672,
  -672, -672, 0, 0, 0, 0, 1344, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 21, 42, 91, 189, 0, 7, 21, -6, -5, -3, 0, 336, 0, 0, 0, 0, 0, 0, 672, 0,
  1344, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
[
  7, 21, 42, 91, 189, 0, 7, 21, -6, -5, -3, 336, 0, 0, 0, 0, 0, 672, 0, 0, 0,
  0, 0, 0, 1344, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
```

```
[
  -2, -3, 24, 10, 3, 192, 94, 45, 24, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, -1344, -1344, -1344, -1344, -1344, 0, 0, 0, 0, 672, 0, 0, 336,
  0, 0
],
[
  -2, -3, 24, 10, 3, 192, 94, 45, 24, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 1344, 0, 0, 672, 0, 0, -336, -336, -336, -336, -336
],
[
  -2, -3, 24, 10, 3, 192, 94, 45, 24, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1344, 0, 672, 0, 0, 0, 0, 0, 336, 0, 0, 0
],
[
  -2, -3, 24, 10, 3, 192, 94, 45, 24, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 1344, 0, 0, -672, -672, -672, -672, -672, 0, 0, 0, 0, 336
],
[
  -2, -3, 24, 10, 3, 192, 94, 45, 24, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 1344, 0, 0, 0, 0, 0, 0, 672, 0, 336, 0, 0, 0, 0
],
[
  -2, -3, 24, 10, 3, 192, 94, 45, 24, 10, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 1344, 0, 0, 0, 0, 0, 672, 0, 0, 0, 0, 0, 0, 336, 0
],
[
  1, 0, 6, 13, 6, 48, 97, 48, 6, 13, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, -336, -336, -336, -336, -336, 0, 0, 0, 0, 672, 0, 0, 336, 0, 0
],
[
  1, 0, 6, 13, 6, 48, 97, 48, 6, 13, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 336, 0, 0, 672, 0, 0, -336, -336, -336, -336, -336
],
[
  1, 0, 6, 13, 6, 48, 97, 48, 6, 13, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 336, 0, 672, 0, 0, 0, 0, 0, 336, 0, 0, 0
],
[
  1, 0, 6, 13, 6, 48, 97, 48, 6, 13, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 336, 0, 0, -672, -672, -672, -672, -672, 0, 0, 0, 0, 336
],
[
  1, 0, 6, 13, 6, 48, 97, 48, 6, 13, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 672, 0, 336, 0, 0, 0, 0
],
[
  1, 0, 6, 13, 6, 48, 97, 48, 6, 13, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 336, 0, 0, 0, 0, 0, 672, 0, 0, 0, 0, 0, 0, 336, 0
],
[
  1, 3, 6, 13, 27, 48, 97, 195, 6, 13, 27, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, -336, -336, -336, -336, -336, 0, 0, 0, 0, 672, 0, 0, 1344, 0, 0
],
[
  1, 3, 6, 13, 27, 48, 97, 195, 6, 13, 27, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 672, 0, 0, -1344, -1344, -1344, -1344, -1344
],
[
  1, 3, 6, 13, 27, 48, 97, 195, 6, 13, 27, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
        0, 0, 0, 0, 0, 0, 336, 0, 672, 0, 0, 0, 0, 0, 1344, 0, 0, 0
      ],
      [
        1, 3, 6, 13, 27, 48, 97, 195, 6, 13, 27, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 336, 0, 0, -672, -672, -672, -672, -672, 0, 0, 0, 0, 1344
      ],
      [
        1, 3, 6, 13, 27, 48, 97, 195, 6, 13, 27, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 672, 0, 1344, 0, 0, 0, 0
      ],
      [
        1, 3, 6, 13, 27, 48, 97, 195, 6, 13, 27, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 336, 0, 0, 0, 0, 0, 672, 0, 0, 0, 0, 0, 0, 1344, 0
      ],
      [
        -2, -3, 24, 10, 3, 192, 94, 45, 1368, 682, 339, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ],
      [
        1, 0, 6, 13, 6, 48, 97, 48, 342, 685, 342, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ],
      [
        1, 3, 6, 13, 27, 48, 97, 195, 342, 685, 1371, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
      ]
    ],
    "deleted_row": 41,
    "square_minor_determinant": {
      "sign": 1,
      "decimal_blocks_to_concatenate": [
        "858640154684361302653112973863117244469725999626552133825583",
        "29297776143170519455739016546146230451111701289361204681113 6"
      ]
    },
    "rank": 41
  },
  {
    "p": 11,
    "n": 3,
    "M": 1,
    "rows": [
      [0, 1, null, 1, 1],
      [1, 1, 1, 1, 11],
      [1, 1, 2, 2, 11],
      [1, 1, 3, 3, 11],
      [1, 1, 4, 4, 11],
      [1, 1, 5, 5, 11],
      [1, 1, 6, 6, 11],
      [1, 1, 7, 7, 11],
      [1, 1, 8, 8, 11],
      [1, 1, 9, 9, 11],
      [1, 1, 10, 10, 11],
      [2, 1, 1, 1, 121],
      [2, 1, 2, 2, 121],
      [2, 1, 3, 3, 121],
      [2, 1, 4, 4, 121],
      [2, 1, 5, 5, 121],
      [2, 1, 6, 6, 121],
```

```
  [2, 1, 7, 7, 121],
  [2, 1, 8, 8, 121],
  [2, 1, 9, 9, 121],
  [2, 1, 10, 10, 121],
  [3, 1, null, 1, 1331]
],
"columns": [
  ["ordinary", 11, null],
  ["ordinary", 121, null],
  ["ordinary", 1331, null],
  ["shift", 1, 1],
  ["shift", 1, 2],
  ["shift", 1, 3],
  ["shift", 1, 4],
  ["shift", 1, 5],
  ["shift", 1, 6],
  ["shift", 1, 7],
  ["shift", 1, 8],
  ["shift", 1, 9],
  ["shift", 11, 1],
  ["shift", 11, 2],
  ["shift", 11, 3],
  ["shift", 11, 4],
  ["shift", 11, 5],
  ["shift", 11, 6],
  ["shift", 11, 7],
  ["shift", 11, 8],
  ["shift", 11, 9]
],
"matrix": [
  [-1210, -1320, -1330, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [
    110, 0, -10, -1320, -1320, -1320, -1320, -1320, -1320, -1320, -1320, -1320,
    0, 0, 0, 0, 0, 0, 0, 0, 0
  ],
  [110, 0, -10, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [110, 0, -10, 1320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [
    10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1320, -1320, -1320, -1320, -1320,
    -1320, -1320, -1320, -1320
  ],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0],
  [10, 120, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1320, 0, 0, 0, 0, 0, 0, 0, 0],
  [10, 120, 1330, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
],
```

```
      "deleted_row": 21,
      "square_minor_determinant": {
        "sign": 1,
        "decimal_blocks_to_concatenate":
["257916201549169018759029635738966915914137600000000000000000", "000000"]
      },
      "rank": 21
    },
    {
      "p": 13,
      "n": 2,
      "M": 2,
      "rows": [
        [0, 1, null, 1, 1],
        [0, 2, null, 1, 2],
        [1, 1, 1, 1, 13],
        [1, 1, 2, 2, 13],
        [1, 1, 3, 3, 13],
        [1, 1, 4, 4, 13],
        [1, 1, 5, 5, 13],
        [1, 1, 6, 6, 13],
        [1, 1, 7, 7, 13],
        [1, 1, 8, 8, 13],
        [1, 1, 9, 9, 13],
        [1, 1, 10, 10, 13],
        [1, 1, 11, 11, 13],
        [1, 1, 12, 12, 13],
        [1, 2, 1, 15, 26],
        [1, 2, 2, 17, 26],
        [1, 2, 3, 19, 26],
        [1, 2, 4, 21, 26],
        [1, 2, 5, 23, 26],
        [1, 2, 6, 25, 26],
        [1, 2, 7, 1, 26],
        [1, 2, 8, 3, 26],
        [1, 2, 9, 5, 26],
        [1, 2, 10, 7, 26],
        [1, 2, 11, 9, 26],
        [1, 2, 12, 11, 26],
        [2, 1, null, 1, 169],
        [2, 2, null, 1, 338]
      ],
      "columns": [
        ["ordinary", 2, null],
        ["ordinary", 13, null],
        ["ordinary", 26, null],
        ["ordinary", 169, null],
        ["ordinary", 338, null],
        ["shift", 1, 1],
        ["shift", 1, 2],
        ["shift", 1, 3],
        ["shift", 1, 4],
        ["shift", 1, 5],
        ["shift", 1, 6],
        ["shift", 1, 7],
        ["shift", 1, 8],
        ["shift", 1, 9],
        ["shift", 1, 10],
        ["shift", 1, 11],
```

```
    ["shift", 2, 1],
    ["shift", 2, 2],
    ["shift", 2, 3],
    ["shift", 2, 4],
    ["shift", 2, 5],
    ["shift", 2, 6],
    ["shift", 2, 7],
    ["shift", 2, 8],
    ["shift", 2, 9],
    ["shift", 2, 10],
    ["shift", 2, 11]
  ],
  "matrix": [
    [
      -169, -312, -325, -336, -337, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0
    ],
    [
      169, -156, -143, -168, -167, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, -336, -336, -336, -336, -336, -336, -336, -336,
      -336, -336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 168
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0,
      0, 168, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 168,
      0, 0, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 168, 0, 0,
      0, 0, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0,
      0, 0, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, -168, -168, -168,
      -168, -168, -168, -168, -168, -168, -168, -168
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 168, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      168, 0, 0, 0
    ],
    [
      -1, 24, 11, 0, -1, 0, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 168, 0,
```

```
  0, 0, 0, 0
],
[
  -1, 24, 11, 0, -1, 0, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0,
  0, 0, 0, 0
],
[
  -1, 24, 11, 0, -1, 336, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, -168, -168, -168, -168, -168, -168, -168, -168, -168, -168,
  -168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 336
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, 0, 0, 0,
  336, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, 0, 0, 336,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, 0, 336, 0, 0,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, -336, -336, -336, -336,
  -336, -336, -336, -336, -336, -336, -336
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 0, 168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 336, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 0, 168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  336, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 0, 168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 336, 0,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 0, 168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0,
  0, 0, 0, 0
],
[
  1, 12, 25, 0, 1, 168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 336, 0, 0, 0, 0, 0,
  0, 0, 0, 0
],
[
```

```
          -1, 24, 11, 336, 167, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0
        ],
        [
          1, 12, 25, 168, 337, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0
        ]
      ],
      "deleted_row": 27,
      "square_minor_determinant": {
        "sign": -1,
        "decimal_blocks_to_concatenate":
["153743033174456242476193344316610720180800725731790147989127", "4039296"]
      },
      "rank": 27
    }
  ],
  "ordinary_norm_stencil": [
    {
      "p": 5,
      "target_depth": 2,
      "matrix": [[6, 5, 0, 0], [-1, 0, 5, 0], [0, 0, 0, 5]],
      "smith_diagonal": [1, 5, 5],
      "congruence_kernel_basis": [[5, -1, 0], [0, 1, 0], [0, 0, 5]],
      "integral_preimage_columns": [[0, 1, 0, 0], [-1, 1, 0, 0], [0, 0, 0, 1]]
    },
    {
      "p": 5,
      "target_depth": 3,
      "matrix": [[6, 5, 0, 0, 0], [-1, 0, 5, 0, 0], [0, 0, 0, 5, 0], [0, 0, 0, 0, 5]],
      "smith_diagonal": [1, 5, 5, 5],
      "congruence_kernel_basis": [[5, -1, 0, 0], [0, 1, 0, 0], [0, 0, 5, 0], [0, 0, 0, 5]],
      "integral_preimage_columns": [[0, 1, 0, 0, 0], [-1, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0,
0, 0, 0, 1]]
    },
    {
      "p": 5,
      "target_depth": 4,
      "matrix": [
        [6, 5, 0, 0, 0, 0],
        [-1, 0, 5, 0, 0, 0],
        [0, 0, 0, 5, 0, 0],
        [0, 0, 0, 0, 5, 0],
        [0, 0, 0, 0, 0, 5]
      ],
      "smith_diagonal": [1, 5, 5, 5, 5],
      "congruence_kernel_basis": [
        [5, -1, 0, 0, 0],
        [0, 1, 0, 0, 0],
        [0, 0, 5, 0, 0],
        [0, 0, 0, 5, 0],
        [0, 0, 0, 0, 5]
      ],
      "integral_preimage_columns": [
        [0, 1, 0, 0, 0, 0],
        [-1, 1, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 1]
```

```
    ]
  },
  {
    "p": 5,
    "target_depth": 5,
    "matrix": [
      [6, 5, 0, 0, 0, 0, 0],
      [-1, 0, 5, 0, 0, 0, 0],
      [0, 0, 0, 5, 0, 0, 0],
      [0, 0, 0, 0, 5, 0, 0],
      [0, 0, 0, 0, 0, 5, 0],
      [0, 0, 0, 0, 0, 0, 5]
    ],
    "smith_diagonal": [1, 5, 5, 5, 5, 5],
    "congruence_kernel_basis": [
      [5, -1, 0, 0, 0, 0],
      [0, 1, 0, 0, 0, 0],
      [0, 0, 5, 0, 0, 0],
      [0, 0, 0, 5, 0, 0],
      [0, 0, 0, 0, 5, 0],
      [0, 0, 0, 0, 0, 5]
    ],
    "integral_preimage_columns": [
      [0, 1, 0, 0, 0, 0, 0],
      [-1, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ]
  },
  {
    "p": 5,
    "target_depth": 6,
    "matrix": [
      [6, 5, 0, 0, 0, 0, 0, 0],
      [-1, 0, 5, 0, 0, 0, 0, 0],
      [0, 0, 0, 5, 0, 0, 0, 0],
      [0, 0, 0, 0, 5, 0, 0, 0],
      [0, 0, 0, 0, 0, 5, 0, 0],
      [0, 0, 0, 0, 0, 0, 5, 0],
      [0, 0, 0, 0, 0, 0, 0, 5]
    ],
    "smith_diagonal": [1, 5, 5, 5, 5, 5, 5],
    "congruence_kernel_basis": [
      [5, -1, 0, 0, 0, 0, 0],
      [0, 1, 0, 0, 0, 0, 0],
      [0, 0, 5, 0, 0, 0, 0],
      [0, 0, 0, 5, 0, 0, 0],
      [0, 0, 0, 0, 5, 0, 0],
      [0, 0, 0, 0, 0, 5, 0],
      [0, 0, 0, 0, 0, 0, 5]
    ],
    "integral_preimage_columns": [
      [0, 1, 0, 0, 0, 0, 0, 0],
      [-1, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0],
```

```
        [0, 0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 0, 0, 1]
      ]
    },
    {
      "p": 7,
      "target_depth": 2,
      "matrix": [[8, 7, 0, 0], [-1, 0, 7, 0], [0, 0, 0, 7]],
      "smith_diagonal": [1, 7, 7],
      "congruence_kernel_basis": [[7, -1, 0], [0, 1, 0], [0, 0, 7]],
      "integral_preimage_columns": [[0, 1, 0, 0], [-1, 1, 0, 0], [0, 0, 0, 1]]
    },
    {
      "p": 7,
      "target_depth": 3,
      "matrix": [[8, 7, 0, 0, 0], [-1, 0, 7, 0, 0], [0, 0, 0, 7, 0], [0, 0, 0, 0, 7]],
      "smith_diagonal": [1, 7, 7, 7],
      "congruence_kernel_basis": [[7, -1, 0, 0], [0, 1, 0, 0], [0, 0, 7, 0], [0, 0, 0, 7]],
      "integral_preimage_columns": [[0, 1, 0, 0, 0], [-1, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0,
0, 0, 0, 1]]
    },
    {
      "p": 7,
      "target_depth": 4,
      "matrix": [
        [8, 7, 0, 0, 0, 0],
        [-1, 0, 7, 0, 0, 0],
        [0, 0, 0, 7, 0, 0],
        [0, 0, 0, 0, 7, 0],
        [0, 0, 0, 0, 0, 7]
      ],
      "smith_diagonal": [1, 7, 7, 7, 7],
      "congruence_kernel_basis": [
        [7, -1, 0, 0, 0],
        [0, 1, 0, 0, 0],
        [0, 0, 7, 0, 0],
        [0, 0, 0, 7, 0],
        [0, 0, 0, 0, 7]
      ],
      "integral_preimage_columns": [
        [0, 1, 0, 0, 0, 0],
        [-1, 1, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 1]
      ]
    },
    {
      "p": 7,
      "target_depth": 5,
      "matrix": [
        [8, 7, 0, 0, 0, 0, 0],
        [-1, 0, 7, 0, 0, 0, 0],
        [0, 0, 0, 7, 0, 0, 0],
        [0, 0, 0, 0, 7, 0, 0],
        [0, 0, 0, 0, 0, 7, 0],
        [0, 0, 0, 0, 0, 0, 7]
      ],
      "smith_diagonal": [1, 7, 7, 7, 7, 7],
```

```
    "congruence_kernel_basis": [
      [7, -1, 0, 0, 0, 0],
      [0, 1, 0, 0, 0, 0],
      [0, 0, 7, 0, 0, 0],
      [0, 0, 0, 7, 0, 0],
      [0, 0, 0, 0, 7, 0],
      [0, 0, 0, 0, 0, 7]
    ],
    "integral_preimage_columns": [
      [0, 1, 0, 0, 0, 0, 0],
      [-1, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ]
  },
  {
    "p": 7,
    "target_depth": 6,
    "matrix": [
      [8, 7, 0, 0, 0, 0, 0, 0],
      [-1, 0, 7, 0, 0, 0, 0, 0],
      [0, 0, 0, 7, 0, 0, 0, 0],
      [0, 0, 0, 0, 7, 0, 0, 0],
      [0, 0, 0, 0, 0, 7, 0, 0],
      [0, 0, 0, 0, 0, 0, 7, 0],
      [0, 0, 0, 0, 0, 0, 0, 7]
    ],
    "smith_diagonal": [1, 7, 7, 7, 7, 7, 7],
    "congruence_kernel_basis": [
      [7, -1, 0, 0, 0, 0, 0],
      [0, 1, 0, 0, 0, 0, 0],
      [0, 0, 7, 0, 0, 0, 0],
      [0, 0, 0, 7, 0, 0, 0],
      [0, 0, 0, 0, 7, 0, 0],
      [0, 0, 0, 0, 0, 7, 0],
      [0, 0, 0, 0, 0, 0, 7]
    ],
    "integral_preimage_columns": [
      [0, 1, 0, 0, 0, 0, 0, 0],
      [-1, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 1]
    ]
  },
  {
    "p": 11,
    "target_depth": 2,
    "matrix": [[12, 11, 0, 0], [-1, 0, 11, 0], [0, 0, 0, 11]],
    "smith_diagonal": [1, 11, 11],
    "congruence_kernel_basis": [[11, -1, 0], [0, 1, 0], [0, 0, 11]],
    "integral_preimage_columns": [[0, 1, 0, 0], [-1, 1, 0, 0], [0, 0, 0, 1]]
  },
  {
    "p": 11,
```

```
      "target_depth": 3,
      "matrix": [[12, 11, 0, 0, 0], [-1, 0, 11, 0, 0], [0, 0, 0, 11, 0], [0, 0, 0, 0, 11]],
      "smith_diagonal": [1, 11, 11, 11],
      "congruence_kernel_basis": [[11, -1, 0, 0], [0, 1, 0, 0], [0, 0, 11, 0], [0, 0, 0,
11]],
      "integral_preimage_columns": [[0, 1, 0, 0, 0], [-1, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0,
0, 0, 0, 1]]
    },
    {
      "p": 11,
      "target_depth": 4,
      "matrix": [
        [12, 11, 0, 0, 0, 0],
        [-1, 0, 11, 0, 0, 0],
        [0, 0, 0, 11, 0, 0],
        [0, 0, 0, 0, 11, 0],
        [0, 0, 0, 0, 0, 11]
      ],
      "smith_diagonal": [1, 11, 11, 11, 11],
      "congruence_kernel_basis": [
        [11, -1, 0, 0, 0],
        [0, 1, 0, 0, 0],
        [0, 0, 11, 0, 0],
        [0, 0, 0, 11, 0],
        [0, 0, 0, 0, 11]
      ],
      "integral_preimage_columns": [
        [0, 1, 0, 0, 0, 0],
        [-1, 1, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 1]
      ]
    },
    {
      "p": 11,
      "target_depth": 5,
      "matrix": [
        [12, 11, 0, 0, 0, 0, 0],
        [-1, 0, 11, 0, 0, 0, 0],
        [0, 0, 0, 11, 0, 0, 0],
        [0, 0, 0, 0, 11, 0, 0],
        [0, 0, 0, 0, 0, 11, 0],
        [0, 0, 0, 0, 0, 0, 11]
      ],
      "smith_diagonal": [1, 11, 11, 11, 11, 11],
      "congruence_kernel_basis": [
        [11, -1, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0],
        [0, 0, 11, 0, 0, 0],
        [0, 0, 0, 11, 0, 0],
        [0, 0, 0, 0, 11, 0],
        [0, 0, 0, 0, 0, 11]
      ],
      "integral_preimage_columns": [
        [0, 1, 0, 0, 0, 0, 0],
        [-1, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 1, 0, 0],
```

```
        [0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 0, 1]
      ]
    },
    {
      "p": 11,
      "target_depth": 6,
      "matrix": [
        [12, 11, 0, 0, 0, 0, 0, 0],
        [-1, 0, 11, 0, 0, 0, 0, 0],
        [0, 0, 0, 11, 0, 0, 0, 0],
        [0, 0, 0, 0, 11, 0, 0, 0],
        [0, 0, 0, 0, 0, 11, 0, 0],
        [0, 0, 0, 0, 0, 0, 11, 0],
        [0, 0, 0, 0, 0, 0, 0, 11]
      ],
      "smith_diagonal": [1, 11, 11, 11, 11, 11, 11],
      "congruence_kernel_basis": [
        [11, -1, 0, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0, 0],
        [0, 0, 11, 0, 0, 0, 0],
        [0, 0, 0, 11, 0, 0, 0],
        [0, 0, 0, 0, 11, 0, 0],
        [0, 0, 0, 0, 0, 11, 0],
        [0, 0, 0, 0, 0, 0, 11]
      ],
      "integral_preimage_columns": [
        [0, 1, 0, 0, 0, 0, 0, 0],
        [-1, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 0, 0, 1]
      ]
    },
    {
      "p": 13,
      "target_depth": 2,
      "matrix": [[14, 13, 0, 0], [-1, 0, 13, 0], [0, 0, 0, 13]],
      "smith_diagonal": [1, 13, 13],
      "congruence_kernel_basis": [[13, -1, 0], [0, 1, 0], [0, 0, 13]],
      "integral_preimage_columns": [[0, 1, 0, 0], [-1, 1, 0, 0], [0, 0, 0, 1]]
    },
    {
      "p": 13,
      "target_depth": 3,
      "matrix": [[14, 13, 0, 0, 0], [-1, 0, 13, 0, 0], [0, 0, 0, 13, 0], [0, 0, 0, 0, 13]],
      "smith_diagonal": [1, 13, 13, 13],
      "congruence_kernel_basis": [[13, -1, 0, 0], [0, 1, 0, 0], [0, 0, 13, 0], [0, 0, 0,
13]],
      "integral_preimage_columns": [[0, 1, 0, 0, 0], [-1, 1, 0, 0, 0], [0, 0, 0, 1, 0], [0,
0, 0, 0, 1]]
    },
    {
      "p": 13,
      "target_depth": 4,
      "matrix": [
        [14, 13, 0, 0, 0, 0],
```

```
      [-1, 0, 13, 0, 0, 0],
      [0, 0, 0, 13, 0, 0],
      [0, 0, 0, 0, 13, 0],
      [0, 0, 0, 0, 0, 13]
    ],
    "smith_diagonal": [1, 13, 13, 13, 13],
    "congruence_kernel_basis": [
      [13, -1, 0, 0, 0],
      [0, 1, 0, 0, 0],
      [0, 0, 13, 0, 0],
      [0, 0, 0, 13, 0],
      [0, 0, 0, 0, 13]
    ],
    "integral_preimage_columns": [
      [0, 1, 0, 0, 0, 0],
      [-1, 1, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 1]
    ]
  },
  {
    "p": 13,
    "target_depth": 5,
    "matrix": [
      [14, 13, 0, 0, 0, 0, 0],
      [-1, 0, 13, 0, 0, 0, 0],
      [0, 0, 0, 13, 0, 0, 0],
      [0, 0, 0, 0, 13, 0, 0],
      [0, 0, 0, 0, 0, 13, 0],
      [0, 0, 0, 0, 0, 0, 13]
    ],
    "smith_diagonal": [1, 13, 13, 13, 13, 13],
    "congruence_kernel_basis": [
      [13, -1, 0, 0, 0, 0],
      [0, 1, 0, 0, 0, 0],
      [0, 0, 13, 0, 0, 0],
      [0, 0, 0, 13, 0, 0],
      [0, 0, 0, 0, 13, 0],
      [0, 0, 0, 0, 0, 13]
    ],
    "integral_preimage_columns": [
      [0, 1, 0, 0, 0, 0, 0],
      [-1, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ]
  },
  {
    "p": 13,
    "target_depth": 6,
    "matrix": [
      [14, 13, 0, 0, 0, 0, 0, 0],
      [-1, 0, 13, 0, 0, 0, 0, 0],
      [0, 0, 0, 13, 0, 0, 0, 0],
      [0, 0, 0, 0, 13, 0, 0, 0],
      [0, 0, 0, 0, 0, 13, 0, 0],
```

```
        [0, 0, 0, 0, 0, 0, 13, 0],
        [0, 0, 0, 0, 0, 0, 0, 13]
      ],
      "smith_diagonal": [1, 13, 13, 13, 13, 13, 13],
      "congruence_kernel_basis": [
        [13, -1, 0, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0, 0],
        [0, 0, 13, 0, 0, 0, 0],
        [0, 0, 0, 13, 0, 0, 0],
        [0, 0, 0, 0, 13, 0, 0],
        [0, 0, 0, 0, 0, 13, 0],
        [0, 0, 0, 0, 0, 0, 13]
      ],
      "integral_preimage_columns": [
        [0, 1, 0, 0, 0, 0, 0, 0],
        [-1, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 0, 0, 1]
      ]
    }
  ],
  "atomic_euler_norm_resultants": [
    {"p": 5, "k": 0, "r": 1, "residual_polynomial": "0"},
    {"p": 5, "k": 0, "r": 2, "residual_polynomial": "0"},
    {"p": 5, "k": 0, "r": 5, "residual_polynomial": "0"},
    {"p": 5, "k": 0, "r": 10, "residual_polynomial": "0"},
    {"p": 5, "k": 1, "r": 1, "residual_polynomial": "0"},
    {"p": 5, "k": 1, "r": 2, "residual_polynomial": "0"},
    {"p": 5, "k": 1, "r": 5, "residual_polynomial": "0"},
    {"p": 5, "k": 1, "r": 10, "residual_polynomial": "0"},
    {"p": 5, "k": 2, "r": 1, "residual_polynomial": "0"},
    {"p": 5, "k": 2, "r": 2, "residual_polynomial": "0"},
    {"p": 5, "k": 2, "r": 5, "residual_polynomial": "0"},
    {"p": 5, "k": 2, "r": 10, "residual_polynomial": "0"},
    {"p": 5, "k": 3, "r": 1, "residual_polynomial": "0"},
    {"p": 5, "k": 3, "r": 2, "residual_polynomial": "0"},
    {"p": 5, "k": 3, "r": 5, "residual_polynomial": "0"},
    {"p": 5, "k": 3, "r": 10, "residual_polynomial": "0"},
    {"p": 5, "k": 4, "r": 1, "residual_polynomial": "0"},
    {"p": 5, "k": 4, "r": 2, "residual_polynomial": "0"},
    {"p": 5, "k": 4, "r": 5, "residual_polynomial": "0"},
    {"p": 5, "k": 4, "r": 10, "residual_polynomial": "0"},
    {"p": 7, "k": 0, "r": 1, "residual_polynomial": "0"},
    {"p": 7, "k": 0, "r": 2, "residual_polynomial": "0"},
    {"p": 7, "k": 0, "r": 7, "residual_polynomial": "0"},
    {"p": 7, "k": 0, "r": 14, "residual_polynomial": "0"},
    {"p": 7, "k": 1, "r": 1, "residual_polynomial": "0"},
    {"p": 7, "k": 1, "r": 2, "residual_polynomial": "0"},
    {"p": 7, "k": 1, "r": 7, "residual_polynomial": "0"},
    {"p": 7, "k": 1, "r": 14, "residual_polynomial": "0"},
    {"p": 7, "k": 2, "r": 1, "residual_polynomial": "0"},
    {"p": 7, "k": 2, "r": 2, "residual_polynomial": "0"},
    {"p": 7, "k": 2, "r": 7, "residual_polynomial": "0"},
    {"p": 7, "k": 2, "r": 14, "residual_polynomial": "0"},
    {"p": 7, "k": 3, "r": 1, "residual_polynomial": "0"},
    {"p": 7, "k": 3, "r": 2, "residual_polynomial": "0"},
```

```
      {"p": 7, "k": 3, "r": 7, "residual_polynomial": "0"},
      {"p": 7, "k": 3, "r": 14, "residual_polynomial": "0"},
      {"p": 7, "k": 4, "r": 1, "residual_polynomial": "0"},
      {"p": 7, "k": 4, "r": 2, "residual_polynomial": "0"},
      {"p": 7, "k": 4, "r": 7, "residual_polynomial": "0"},
      {"p": 7, "k": 4, "r": 14, "residual_polynomial": "0"},
      {"p": 7, "k": 5, "r": 1, "residual_polynomial": "0"},
      {"p": 7, "k": 5, "r": 2, "residual_polynomial": "0"},
      {"p": 7, "k": 5, "r": 7, "residual_polynomial": "0"},
      {"p": 7, "k": 5, "r": 14, "residual_polynomial": "0"},
      {"p": 7, "k": 6, "r": 1, "residual_polynomial": "0"},
      {"p": 7, "k": 6, "r": 2, "residual_polynomial": "0"},
      {"p": 7, "k": 6, "r": 7, "residual_polynomial": "0"},
      {"p": 7, "k": 6, "r": 14, "residual_polynomial": "0"}
    ],
    "first_moment_counterexample": {
      "all_p_ge_5": true,
      "shift_exponents": [2, -1],
      "first_moment": 0,
      "full_polynomial": "2(X-1)-(X^2-1)=-(X-1)^2",
      "nonzero_mod_Phi_p_mod_p": true,
      "interpretation": "The first pi moment vanishes while some shifted exponents are nonzero
   modulo p."
    }
  },
  "all_assertions_passed": true,
  "script_sha256": "62b95fda3fc3d247a07704c1e93dbefc009b426724c909f3831940cef76a35a0"
}
```

A.4. **Ternary geometry identities.** The full symbolic endpoint matrices, their commutator and divisibility quotients, the three nilpotent coefficients, the conductor identity, and the shallow square-zero identity. The symbolic parameter is arbitrary. The local calculations are (8.73), (8.67), and (8.74).

Program.

```
"""Exact symbolic identities for the ternary geometry calculations.

Run with Python 3 and SymPy.  All arithmetic below is exact.  The parameter m
stands for N/27, so the matrix identities hold at every depth n >= 3.
"""
import json
from pathlib import Path
import sympy as s

m = s.symbols('m', integer=True)
N = 27*m
S = s.Matrix([[1, 0], [-4*N/3, 1]])
T = s.Matrix([[1, s.Rational(4, 3)], [0, 1]])
A = s.diag(3, 1)
comm = s.simplify(S*T*S.inv()*T.inv())
checks = {}
checks['S_cubed'] = S**3 == s.Matrix([[1, 0], [-4*N, 1]])
checks['T_cubed'] = T**3 == s.Matrix([[1, 4], [0, 1]])
checks['common_lower_S'] = A*S*A.inv() == s.Matrix([[1, 0], [-4*N/9, 1]])
checks['common_lower_T'] = A*T*A.inv() == s.Matrix([[1, 4], [0, 1]])
entries = [(comm[0, 0]-1)/4, comm[0, 1]/4,
           comm[1, 0]/N, (comm[1, 1]-1)/4]
checks['commutator_in_Gamma0N_and_Gamma4'] = all(
    s.Poly(s.expand(x), m).domain == s.ZZ for x in entries)

a0,a1,a2 = s.symbols('a0 a1 a2')
v = [a1-a0, a2-a1, a0-a2]
checks['normalized_first_coefficients_sum_zero'] = s.expand(sum(v)) == 0
nu = []
for k in range(3):
    shifted = v[k:] + v[:k]
    nu.append(s.expand(shifted[2]-shifted[1]))
checks['three_original_nil_coefficients_equal_mod_3'] = all(
    s.Poly(s.expand(x-nu[0]), a0,a1,a2, modulus=3).is_zero for x in nu)

# A uniformizer lambda' with (lambda')^2 = -3 is used in the geometric proof.
# In B=A0[w]/(w^2-xy), lambda'=G w.  The following is the exact conductor
# calculation retaining the supersingular factor G.
x,y,G,a,b,w = s.symbols('x y G a b w')
conductor = s.rem(G*w*(a+b*w) - (G*w*a + G*x*y*b), w*w-x*y, w)
checks['full_conductor_identity'] = s.expand(conductor) == 0

# The shallow-square-zero ideal J=I_(F-C_L-C_R)/I_F has generator
# h=G0 tL tR when F=G0 tL^2 tR^2.  h^2=G0 F.
g0,tL,tR = s.symbols('g0 tL tR')
checks['paired_shallow_square_zero'] = s.expand(
    (g0*tL*tR)**2 - g0*(g0*tL*tL*tR*tR)) == 0

assert all(checks.values()), checks
result = {'exact_checks':checks,
          'commutator':[[str(z) for z in row] for row in comm.tolist()],
          'commutator_integral_divisibility_quotients':[str(s.expand(z)) for z in entries],
```

```
          'three_nil_coefficients':[str(z) for z in nu],
          'scope':'Symbolic identities for arbitrary integer m=N/27.'}
destination = Path(__file__).with_suffix('.json')
destination.write_text(json.dumps(result, indent=2), encoding='utf-8')
print(json.dumps(result, indent=2))
```

Output.


```
{
  "exact_checks": {
    "S_cubed": true,
    "T_cubed": true,
    "common_lower_S": true,
    "common_lower_T": true,
    "commutator_in_Gamma0N_and_Gamma4": true,
    "normalized_first_coefficients_sum_zero": true,
    "three_original_nil_coefficients_equal_mod_3": true,
    "full_conductor_identity": true,
    "paired_shallow_square_zero": true
  },
  "commutator": [["48*m + 1", "-64*m"], ["-1728*m**2", "48*m*(48*m - 1) + 1"]],
  "commutator_integral_divisibility_quotients": ["12*m", "-16*m", "-64*m", "576*m**2 - 12*m"],
  "three_nil_coefficients": ["a0 + a1 - 2*a2", "-2*a0 + a1 + a2", "a0 - 2*a1 + a2"],
  "scope": "Symbolic identities for arbitrary integer m=N/27."
}
```

A.5. **Ternary Prym and auxiliary-prime identities.** The cubic quotient, full-level-six map, quartic good model, semilinear inertia and cubic action, all projective full-level-four group elements and stabilizers, every coefficient of the A_4 identity, and the full auxiliary-five packet matrices. These are the calculations in Proposition 8.5 and (8.20)–(8.21).

Program.

```
"""Exact formula certificates for the cubic/Prym identities, 26 November 2026.

Python 3 and SymPy compute the displayed polynomial identities and
finite-group coefficients exactly.
"""
from pathlib import Path
import itertools
import json
import sympy as s

out = {}
lam, x, y, f, r, pi, u, v, t, z, ii = s.symbols(
    'lam x y f r pi u v t z ii')
j_leg = 256*(lam**2-lam+1)**3/(lam**2*(1-lam)**2)
w = 16*(lam**2-lam+1)/x**2
num = s.together(w**3-j_leg).as_numer_denom()[0]
assert s.rem(num, x**3-4*lam*(lam-1), x) == 0
inverse_residual = s.together(lam*(lam-1)*w/(4*(lam**2-lam+1))-x)
assert s.rem(inverse_residual.as_numer_denom()[0],
             x**3-4*lam*(lam-1), x) == 0
out['legendre_cube_and_inverse'] = [0, 0]

lambda_k = (r+1)*(r-3)**3/((r-1)*(r+3)**3)
y_double = (r**4+18*r**2-27)/(8*r**3)
assert s.cancel((y_double+1)/2-1/(1-lambda_k)) == 0
x_double = (f**4-8*f)/(4*(f**3+1))
y_double_f = (f**6+20*f**3-8)/(8*r**3)
double_num = s.together(y_double_f**2-x_double**3-1).as_numer_denom()[0]
assert s.rem(double_num, r**2-f**3-1, r) == 0
out['actual_full6_doubling'] = [0, 0]

good_eq = v**2-u**3+pi**2*u**2-u
old_eq = (pi**3*v)**2-(-1+pi**2*u)**3-1
assert s.rem(s.expand(old_eq-pi**6*good_eq), pi**4-3, pi) == 0
infinity_eq = s.cancel(z**3*good_eq.subs({u:t/z, v:-1/z}))
assert s.expand(infinity_eq-(z-t**3+pi**2*t**2*z-t*z**2)) == 0
assert s.Poly(s.diff(good_eq.subs(pi,0),u),u, modulus=3).as_expr() == -1
out['quartic_good_model_and_infinity'] = [0, 0, -1]

# Semilinear inertia: substitute the uniformizer as well as coordinates.
sigma = {pi:ii*pi, u:-u, v:ii*v}
sigma_res = s.expand(good_eq.xreplace(sigma)+good_eq)
assert s.rem(sigma_res, ii**2+1, ii) == 0
zeta = (-1+ii*pi**2)/2
rho_u = zeta*u+(pi**2-ii)/2
rho_res = s.expand(good_eq.subs(u,rho_u)-good_eq)
rho_res = s.rem(s.rem(rho_res, ii**2+1, ii), pi**4-3, pi)
assert s.expand(rho_res) == 0
assert s.rem(s.expand(zeta**2+zeta+1), ii**2+1, ii).subs(pi**4,3) == 0
out['sigma_and_rho_good_model'] = [0, 0]

# The full mod-4 projective deck group and the even-M stabilizer.
```

```
def mm(a,b,n):
    return tuple(sum(a[2*i+k]*b[2*k+j] for k in range(2))%n
                 for i in range(2) for j in range(2))
def canonical(a,n):
    return min(a, tuple((-c)%n for c in a))
def projective_sl2(n):
    return sorted({canonical(a,n) for a in itertools.product(range(n),repeat=4)
                   if (a[0]*a[3]-a[1]*a[2])%n==1})
group4 = projective_sl2(4)
ident4 = canonical((1,0,0,1),4)
mul4 = lambda a,b: canonical(mm(a,b,4),4)
inv4 = {a:next(b for b in group4 if mul4(a,b)==ident4) for a in group4}
generators = {mul4(mul4(mul4(a,b),inv4[a]),inv4[b])
              for a in group4 for b in group4}
derived = {ident4}
while True:
    enlarged = derived | {mul4(a,b) for a in derived for b in generators}
    if enlarged == derived:
        break
    derived = enlarged
even_stabilizer = [a for a in group4 if a[2]%2==0]
assert len(group4)==24 and len(derived)==12 and len(even_stabilizer)==8
out['full4_deck_group'] = {
    'elements':group4, 'derived_subgroup':sorted(derived),
    'abelianization_order':2, 'even_M_stabilizer':even_stabilizer}

# The complete coefficientwise A4 identity, using its action on four letters.
perms = list(itertools.permutations(range(4)))
def sign(a):
    return (-1)**sum(a[i]>a[j] for i in range(4) for j in range(i+1,4))
def compose(a,b):
    return tuple(a[b[i]] for i in range(4))
ident = tuple(range(4))
group = [a for a in perms if sign(a)==1]
vfour = [a for a in group if compose(a,a)==ident]
stabilizers = [[a for a in group if a[k]==k] for k in range(4)]
residual = [sum(a in h for h in stabilizers)-1+int(a in vfour)
            -4*int(a==ident) for a in group]
assert residual == [0]*12
out['a4_group_ring'] = {'group':group,'vfour':vfour,
                        'stabilizers':stabilizers,'residual':residual}

rr=s.symbols('rr')
assert s.expand((1-rr)**2+3*rr-(1+rr+rr**2)) == 0
out['prym_delta_identity'] = 0

# Entire packet maps, including the cyclotomic permutation.
I=s.eye(2)
tau=s.Matrix([[0,1],[1,0]])
zero=s.zeros(2)
alpha=(5*I).col_join(I)
beta=tau.col_join(5*tau)
W=zero.row_join(tau).col_join(tau.row_join(zero))
Tau=tau.row_join(zero).col_join(zero.row_join(tau))
plus=alpha+beta
minus=alpha-beta
anti=s.Matrix([1,-1])
rat=s.Matrix([1,1])
assert W*alpha==beta and W*beta==alpha
```

```
assert W*plus==plus and W*minus==-minus
assert plus*anti==4*anti.col_join(-anti)
assert minus*anti==6*anti.col_join(anti)
assert minus*rat==4*rat.col_join(-rat)
projector=(s.eye(4)-Tau)*(s.eye(4)+W)/4
assert projector**2==projector and projector.rank()==1
assert projector*plus*anti==plus*anti
out['auxiliary5_packets']={
    'alpha':alpha.tolist(),'beta':beta.tolist(),'w5':W.tolist(),
    'tau':Tau.tolist(),'plus_on_anti':list(plus*anti),
    'minus_on_anti':list(minus*anti),
    'minus_on_rational':list(minus*rat),
    'simultaneous_projector':projector.tolist()}

# Exact anti weights and valuation thresholds; finite residue calculation.
weights = {chi:[r for r in range(4) if (r-chi-2)%4==0]
           for chi in (0,1)}
assert weights == {0:[2],1:[3]}
out['anti_weights_mod4']={'central':weights[0],'tail':weights[1],
                          'threshold_e_over_pminus1':2,
                          'depth_of_three_times_depth2_at_least':6,
                          'depth_of_delta_at_least':3}

def encode(vv):
    if isinstance(vv,s.Basic):
        return int(vv) if vv.is_Integer else str(vv)
    raise TypeError(type(vv).__name__)

destination=Path(__file__).with_suffix('.json')
destination.write_text(json.dumps(out,ensure_ascii=False,indent=2,default=encode)+'\n')
print('All cubic/Prym exact identities passed.')
print('Projective full-4 group: 24; commutator subgroup: 12; even stabilizer: 8.')
print('A4 identity: 12 zero coefficients; auxiliary plus coefficient: 4; discarded minus
  coefficient: 6.')
print(destination.name)
```

Output.


```
{
  "legendre_cube_and_inverse": [0, 0],
  "actual_full6_doubling": [0, 0],
  "quartic_good_model_and_infinity": [0, 0, -1],
  "sigma_and_rho_good_model": [0, 0],
  "full4_deck_group": {
    "elements": [
      [0, 1, 3, 0],
      [0, 1, 3, 1],
      [0, 1, 3, 2],
      [0, 1, 3, 3],
      [1, 0, 0, 1],
      [1, 0, 1, 1],
      [1, 0, 2, 1],
      [1, 0, 3, 1],
      [1, 1, 0, 1],
      [1, 1, 1, 2],
      [1, 1, 2, 3],
      [1, 1, 3, 0],
      [1, 2, 0, 1],
      [1, 2, 1, 3],
```

```
      [1, 2, 2, 1],
      [1, 2, 3, 3],
      [1, 3, 0, 1],
      [1, 3, 1, 0],
      [1, 3, 2, 3],
      [1, 3, 3, 2],
      [2, 1, 1, 1],
      [2, 1, 1, 3],
      [2, 1, 3, 0],
      [2, 1, 3, 2]
    ],
    "derived_subgroup": [
      [0, 1, 3, 1],
      [0, 1, 3, 3],
      [1, 0, 0, 1],
      [1, 0, 2, 1],
      [1, 1, 1, 2],
      [1, 1, 3, 0],
      [1, 2, 0, 1],
      [1, 2, 2, 1],
      [1, 3, 1, 0],
      [1, 3, 3, 2],
      [2, 1, 1, 1],
      [2, 1, 1, 3]
    ],
    "abelianization_order": 2,
    "even_M_stabilizer": [
      [1, 0, 0, 1],
      [1, 0, 2, 1],
      [1, 1, 0, 1],
      [1, 1, 2, 3],
      [1, 2, 0, 1],
      [1, 2, 2, 1],
      [1, 3, 0, 1],
      [1, 3, 2, 3]
    ]
  },
  "a4_group_ring": {
    "group": [
      [0, 1, 2, 3],
      [0, 2, 3, 1],
      [0, 3, 1, 2],
      [1, 0, 3, 2],
      [1, 2, 0, 3],
      [1, 3, 2, 0],
      [2, 0, 1, 3],
      [2, 1, 3, 0],
      [2, 3, 0, 1],
      [3, 0, 2, 1],
      [3, 1, 0, 2],
      [3, 2, 1, 0]
    ],
    "vfour": [[0, 1, 2, 3], [1, 0, 3, 2], [2, 3, 0, 1], [3, 2, 1, 0]],
    "stabilizers": [
      [[0, 1, 2, 3], [0, 2, 3, 1], [0, 3, 1, 2]],
      [[0, 1, 2, 3], [2, 1, 3, 0], [3, 1, 0, 2]],
      [[0, 1, 2, 3], [1, 3, 2, 0], [3, 0, 2, 1]],
      [[0, 1, 2, 3], [1, 2, 0, 3], [2, 0, 1, 3]]
    ],
```

```
    "residual": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
  },
  "prym_delta_identity": 0,
  "auxiliary5_packets": {
    "alpha": [[5, 0], [0, 5], [1, 0], [0, 1]],
    "beta": [[0, 1], [1, 0], [0, 5], [5, 0]],
    "w5": [[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0]],
    "tau": [[0, 1, 0, 0], [1, 0, 0, 0], [0, 0, 0, 1], [0, 0, 1, 0]],
    "plus_on_anti": [4, -4, -4, 4],
    "minus_on_anti": [6, -6, 6, -6],
    "minus_on_rational": [4, 4, -4, -4],
    "simultaneous_projector": [
      ["1/4", "-1/4", "-1/4", "1/4"],
      ["-1/4", "1/4", "1/4", "-1/4"],
      ["-1/4", "1/4", "1/4", "-1/4"],
      ["1/4", "-1/4", "-1/4", "1/4"]
    ]
  },
  "anti_weights_mod4": {
    "central": [2],
    "tail": [3],
    "threshold_e_over_pminus1": 2,
    "depth_of_three_times_depth2_at_least": 6,
    "depth_of_delta_at_least": 3
  }
}
```

A.6. **Ternary units and complete norm lattices.** The exceptional cubic norms and cancellation, the wild cusp and its complete chord and tangent identities at level 27, all 24 multiplier values and all 576 multiplication residuals, the universal eta stencil and kernel, both shallow corrections, all six boundary residues, and twenty complete integral norm-lattice comparisons. The general formulas are (8.28), (8.29), (8.30), and (8.31). In each lattice record let I be the full norm image matrix and P the predicted basis matrix. The printed integer matrices C, D satisfy $IC = P$ and $PD = I$, which prove both inclusions between their integer column lattices.

Program.

```
"""Complete exact checks for the ternary unit identities.

Python 3 and SymPy.  These are finite formula and lattice certificates;
the all-depth geometric and norm arguments are proved in the article.
Every input matrix, congruence basis, and output HNF is recorded in JSON.
"""
from pathlib import Path
from math import gcd
import json
import sympy as s
from sympy.matrices.normalforms import hermite_normal_form

out = {}
zeta,W,z = s.symbols('zeta W z')
phi = zeta*zeta+zeta+1
def red(f):
    return s.rem(s.expand(f), phi, zeta).expand()
a,b = 3*(zeta-1),3*(zeta*zeta-1)
rho = zeta-zeta*zeta
disc = red((W-a)*(W-b))
Y = 2*W+9
assert disc == W*W+9*W+27
assert red(Y*Y-4*disc+27) == 0
num_norm = disc-27
den_norm = red((Y-3*rho)**3+(-2*(zeta*zeta-1))**3*disc)
assert red(num_norm-W*(W+9)) == 0
assert red(den_norm-8*(W-a)*W*(W+9)) == 0
assert red(8*num_norm*(W-a)-den_norm) == 0
line = red((3*rho+2*(zeta*zeta-1)*z)**2-4*z**3+27)
assert red(line+4*z*(z-3)*(z-3*zeta)) == 0
assert red(-9-3*rho-6*(zeta*zeta-1)) == 0
out['actual_X027'] = {
    'curve':'Y^2=4*z^3-27; Y=2*W+9',
    'z_cubed':str(disc),'numerator_norm':str(s.factor(num_norm)),
    'denominator_norm':str(den_norm),
    'line_intersection':'-4*z*(z-3)*(z-3*zeta)',
    'norm_of_f':'1/(W-3*(zeta-1))',
    'all_polynomial_residuals':0}

# The cusp at 0 and its cubic deck difference on X_0(27).
x = s.symbols('x')
yu = -2-3*zeta
slope, intercept = zeta*zeta-1, 1+3*zeta
line_y = slope*x+intercept
flex = s.expand((3*x-5)**2+(3*x-5)-x**3+7)
chord = red(line_y**2+line_y-x**3+7)
wild_residuals = {
    'rational_cusp_flex':s.expand(flex+(x-3)**3),
    'deck_difference_on_curve':red(yu**2+yu+7),
```

```
    'conjugate_equals_inverse':red((-2-3*zeta*zeta)-(-1-yu)),
    'chord_at_minus_P0':red(3*slope+intercept+5),
    'chord_at_TP0':red(3*zeta*slope+intercept-4),
    'chord_factorization':red(chord+x*(x-3)*(x-3*zeta)),
    'inverse_of_third_intersection':red(yu+intercept+1),
    'horizontal_flex_at_u':red(yu**2+yu-x**3+7+x**3)}
assert all(value==0 for value in wild_residuals.values())
out['wild_cusp_X027'] = {
    'curve':'y^2+y=x^3-7', 'P0':[3,4],
    'TP0':['3*zeta',4], 'u':['0',str(yu)],
    'slope':str(slope), 'intercept':str(intercept),
    'rational_cusp_tangent_polynomial':str(flex),
    'chord_intersection_polynomial':str(chord),
    'all_named_residuals':{key:int(value) for key,value in wild_residuals.items()}}

# Test the eta^8 multiplier on every group element.
G=[(a,b,c,d) for a in range(3) for b in range(3)
   for c in range(3) for d in range(3) if (a*d-b*c)%3==1]
def mul(g,h):
    a,b,c,d=g; e,f,k,l=h
    return ((a*e+b*k)%3,(a*f+b*l)%3,
            (c*e+d*k)%3,(c*f+d*l)%3)
def chi(g):
    a,b,c,d=g
    return (a*b if c==0 else (a+d)*pow(c,-1,3))%3
residuals=[(chi(mul(g,h))-chi(g)-chi(h))%3 for g in G for h in G]
assert len(G)==24 and residuals==[0]*576
out['eta8_multiplier']={'elements':G,'values':[chi(g) for g in G],
                        'all_576_residuals':residuals}

# Universal source stencil and all shallow recovery exponents.
p = s.symbols('p', integer=True, positive=True)
b0,b1,b2,d0,k0 = s.symbols('b0 b1 b2 d0 k0')
src=s.Matrix([-b1+p*k0,(b0+(p+1)*b1)/p-(p+1)*k0,k0,b2/p])
mat=s.Matrix([[p+1,p,0,0],[-1,0,p,0],[0,0,0,p]])
assert s.simplify(mat*src-s.Matrix([b0,b1,b2]))==s.zeros(3,1)
v=s.Matrix([2*b0-d0,2*b1+4*d0,2*b2-d0])
assert s.expand((s.Matrix([[9,3,1]])*v)[0]-2*(9*b0+3*b1+b2+d0))==0
assert s.expand((s.Matrix([[1,3,9]])*v)[0]-2*(b0+3*b1+9*b2+d0))==0
assert sum([3,-12,3,6])==0 and 3-36+27+6==0
out['shallow_recovery']={'unit_exponents':[3,-12,3,6],
 'ordinary_exponents_of_v':[str(t) for t in v],
 'recovery_4_minus_3':4-3,'recovery_16_minus_15':16-15,
 'reciprocal_and_infinity_residuals':[0,0]}
boundary=[]
for d in [1,2,4,5,7,8]:
    eps=1 if d%3==1 else -1
    r=(4*pow(d,-1,9)+d+eps-6*eps)%9
    assert r==0
    boundary.append([d,pow(d,-1,9),eps,r])
out['six_boundary_classes']=boundary

# Explicit integer congruence intersections.  Each column operation below
# has determinant one; the last scaling imposes exactly one congruence.
def egcd(a,b):
    g,x,y=s.gcdex(s.Integer(a),s.Integer(b))
    return int(y),int(g),int(x)
def impose(B,coeff,modulus):
    B=B.copy(); row=list(map(int,s.Matrix([coeff])*B))
```

```
    for j in range(1,B.cols):
        a,b=row[0],row[j]
        if a==0 and b==0: continue
        g,x,y=egcd(a,b)
        if g<0: g,x,y=-g,-x,-y
        c0,cj=B[:,0],B[:,j]
        B[:,0]=x*c0+y*cj
        B[:,j]=(-b//g)*c0+(a//g)*cj
        row[0],row[j]=g,0
    B[:,0]*=modulus//gcd(modulus,row[0])
    assert all(int(v)%modulus==0 for v in s.Matrix([coeff])*B)
    # Canonical column HNF changes the basis and preserves the integer lattice.
    # Reducing after every intersection avoids artificial coefficient growth.
    return hermite_normal_form(B)
def valuation(d,p):
    v=0
    while d%p==0: d//=p; v+=1
    return v
def eta_lattice(N):
    ds=list(map(int,s.divisors(N))); B=s.zeros(len(ds),len(ds)-1)
    for j in range(B.cols): B[j,j]=1; B[-1,j]=-1
    conditions=[(ds,24),([N//d for d in ds],24)]
    conditions += [([valuation(d,int(p)) for d in ds],2) for p in s.factorint(N)]
    for row,m in conditions: B=impose(B,row,m)
    return ds,B
def ints(A): return [[int(x) for x in row] for row in A.tolist()]

def lattice_witnesses(depth, tame, ds, Ds, H, norm, pred):
    N=3**depth*tame
    columns=[]
    for j in range(pred.cols):
        b=dict(zip(ds,pred[:,j])); a={d:0 for d in Ds}
        for d in map(int,s.divisors(tame)):
            assert (b[d]+4*b[3*d])%3==0
            a[d]=-b[3*d]; a[3*d]=(b[d]+4*b[3*d])//3
            for i in range(2,depth):
                assert b[3**i*d]%3==0
                a[3**(i+1)*d]=b[3**i*d]//3
        reciprocal=sum((N//d)*a[d] for d in Ds)
        z=next(z for z in range(24)
               if (reciprocal+16*3**(depth-2)*z)%24==0)
        a[tame]+=3*z; a[3*tame]-=4*z; a[9*tame]+=z
        av=s.Matrix([a[d] for d in Ds])
        assert sum(av)==0 and norm*av==pred[:,j]
        # Deleting the last coordinate identifies the weight-zero lattice.
        c=H[:-1,:].inv()*av[:-1,:]
        assert all(v.is_Integer for v in c) and H*c==av
        columns.append(c)
    C=s.Matrix.hstack(*columns)
    image=norm*H
    D=pred[:-1,:].inv()*image[:-1,:]
    assert all(v.is_Integer for v in D)
    assert image*C==pred and pred*D==image
    return C,D

cases=[]
for depth in [2,3,4,5]:
    for tame in [1,2,4,5,10]:
        N=3**depth*tame; n=N//3
```

```
        ds,B=eta_lattice(n); Ds,H=eta_lattice(N)
        index={d:i for i,d in enumerate(ds)}
        norm=s.zeros(len(ds),len(Ds))
        for j,d in enumerate(Ds):
            if d%3:
                norm[index[d],j]=4; norm[index[3*d],j]=-1
            else: norm[index[d//3],j]=3
        raw=B
        for d in map(int,s.divisors(tame)):
            row=[0]*len(ds); row[index[d]]=1; row[index[3*d]]=1
            raw=impose(raw,row,3)
            for i in range(2,depth):
                row=[0]*len(ds); row[index[3**i*d]]=1
                raw=impose(raw,row,3)
        pred=raw
        if depth>=3: pred=impose(pred,[n//d for d in ds],72)
        image=norm*H
        left=hermite_normal_form(image[:-1,:])
        right=hermite_normal_form(pred[:-1,:])
        assert left==right,(depth,tame)
        C,D=lattice_witnesses(depth,tame,ds,Ds,H,norm,pred)
        cases.append({'depth':depth,'tame':tame,'lower_divisors':ds,
          'upper_divisors':Ds,'lower_eta_basis':ints(B),
          'upper_eta_basis':ints(H),'norm_matrix':ints(norm),
          'raw_congruence_basis':ints(raw),'predicted_basis':ints(pred),
          'full_norm_image':ints(image),'common_deleted_row_HNF':ints(left),
          'image_preimage_matrix_C':ints(C),
          'predicted_coordinates_of_image_D':ints(D),
          'image_times_C_equals_predicted':True,
          'predicted_times_D_equals_image':True,
          'all_equal':True})
out['complete_lattice_cases']=cases
out['summary']={'lattice_cases':len(cases),'multiplier_pairs':len(residuals),
                'status':'all exact assertions passed'}
Path(__file__).with_suffix('.json').write_text(json.dumps(out,indent=2)+'\n')
print(json.dumps(out['summary']))
```

Output.


```
{
  "actual_X027": {
    "curve": "Y^2=4*z^3-27; Y=2*W+9",
    "z_cubed": "W**2 + 9*W + 27",
    "numerator_norm": "W*(W + 9)",
    "denominator_norm": "8*W**3 - 24*W**2*zeta + 96*W**2 - 216*W*zeta + 216*W",
    "line_intersection": "-4*z*(z-3)*(z-3*zeta)",
    "norm_of_f": "1/(W-3*(zeta-1))",
    "all_polynomial_residuals": 0
  },
  "wild_cusp_X027": {
    "curve": "y^2+y=x^3-7",
    "P0": [3, 4],
    "TP0": ["3*zeta", 4],
    "u": ["0", "-3*zeta - 2"],
    "slope": "zeta**2 - 1",
    "intercept": "3*zeta + 1",
    "rational_cusp_tangent_polynomial": "-x**3 + 9*x**2 - 27*x + 27",
    "chord_intersection_polynomial": "-x**3 + 3*x**2*zeta + 3*x**2 - 9*x*zeta",
    "all_named_residuals": {
```

```
    "rational_cusp_flex": 0,
    "deck_difference_on_curve": 0,
    "conjugate_equals_inverse": 0,
    "chord_at_minus_P0": 0,
    "chord_at_TP0": 0,
    "chord_factorization": 0,
    "inverse_of_third_intersection": 0,
    "horizontal_flex_at_u": 0
  }
},
"eta8_multiplier": {
  "elements": [
    [0, 1, 2, 0],
    [0, 1, 2, 1],
    [0, 1, 2, 2],
    [0, 2, 1, 0],
    [0, 2, 1, 1],
    [0, 2, 1, 2],
    [1, 0, 0, 1],
    [1, 0, 1, 1],
    [1, 0, 2, 1],
    [1, 1, 0, 1],
    [1, 1, 1, 2],
    [1, 1, 2, 0],
    [1, 2, 0, 1],
    [1, 2, 1, 0],
    [1, 2, 2, 2],
    [2, 0, 0, 2],
    [2, 0, 1, 2],
    [2, 0, 2, 2],
    [2, 1, 0, 2],
    [2, 1, 1, 1],
    [2, 1, 2, 0],
    [2, 2, 0, 2],
    [2, 2, 1, 0],
    [2, 2, 2, 1]
  ],
  "values": [0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 0, 2, 2, 1, 0, 0, 1, 2, 2, 0, 1, 1, 2, 0],
  "all_576_residuals": [
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0
  ]
},
"shallow_recovery": {
  "unit_exponents": [3, -12, 3, 6],
  "ordinary_exponents_of_v": ["2*b0 - d0", "2*b1 + 4*d0", "2*b2 - d0"],
  "recovery_4_minus_3": 1,
  "recovery_16_minus_15": 1,
  "reciprocal_and_infinity_residuals": [0, 0]
},
"six_boundary_classes": [
  [1, 1, 1, 0],
  [2, 5, -1, 0],
  [4, 7, 1, 0],
  [5, 2, -1, 0],
  [7, 4, 1, 0],
  [8, 8, -1, 0]
],
"complete_lattice_cases": [
  {
    "depth": 2,
    "tame": 1,
    "lower_divisors": [1, 3],
    "upper_divisors": [1, 3, 9],
    "lower_eta_basis": [[-12], [12]],
    "upper_eta_basis": [[-12, -3], [12, 0], [0, 3]],
    "norm_matrix": [[4, 3, 0], [-1, 0, 3]],
    "raw_congruence_basis": [[-12], [12]],
    "predicted_basis": [[-12], [12]],
    "full_norm_image": [[-12, -12], [12, 12]],
    "common_deleted_row_HNF": [[12]],
    "image_preimage_matrix_C": [[1], [0]],
    "predicted_coordinates_of_image_D": [[1, 1]],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 2,
    "tame": 2,
    "lower_divisors": [1, 2, 3, 6],
    "upper_divisors": [1, 2, 3, 6, 9, 18],
    "lower_eta_basis": [[-24, -12, -13], [24, 0, 5], [0, 12, 7], [0, 0, 1]],
    "upper_eta_basis": [
      [-24, -20, -17, -3, -25],
      [24, 16, 13, 0, 23],
      [0, 4, 3, 0, 0],
      [0, 0, 1, 0, 0],
      [0, 0, 0, 3, 1],
      [0, 0, 0, 0, 1]
    ],
    "norm_matrix": [[4, 0, 3, 0, 0, 0], [0, 4, 0, 3, 0, 0], [-1, 0, 0, 0, 3, 0], [0, -1, 0,
0, 0, 3]],
    "raw_congruence_basis": [[-24, -12, -13], [24, 0, 5], [0, 12, 7], [0, 0, 1]],
    "predicted_basis": [[-24, -12, -13], [24, 0, 5], [0, 12, 7], [0, 0, 1]],
    "full_norm_image": [
      [-96, -68, -59, -12, -100],
      [96, 64, 55, 0, 92],
```

```
      [24, 20, 17, 12, 28],
      [-24, -16, -13, 0, -20]
    ],
    "common_deleted_row_HNF": [[24, 0, 5], [0, 12, 11], [0, 0, 1]],
    "image_preimage_matrix_C": [[1, -2, -1], [-8, 3, -1], [8, 0, 3], [0, 0, 0], [0, 0, 0]],
    "predicted_coordinates_of_image_D": [[9, 6, 5, 0, 8], [16, 11, 9, 1, 14], [-24, -16,
-13, 0, -20]],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 2,
    "tame": 4,
    "lower_divisors": [1, 2, 3, 4, 6, 12],
    "upper_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 36],
    "lower_eta_basis": [
      [-24, -18, -8, -19, -23],
      [24, 12, 0, 17, 12],
      [0, 6, 0, 1, 5],
      [0, 0, 8, 0, 5],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-24, -22, -8, -19, -3, -27, -25, -10],
      [24, 20, 0, 17, 0, 20, 23, 0],
      [0, 2, 0, 1, 0, 1, 0, 0],
      [0, 0, 8, 0, 0, 5, 0, 7],
      [0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 1, 2],
      [0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 0, 3, 0, 0, 0, 0, 0, 0],
      [0, 4, 0, 0, 3, 0, 0, 0, 0],
      [-1, 0, 0, 0, 0, 3, 0, 0, 0],
      [0, 0, 0, 4, 0, 0, 3, 0, 0],
      [0, -1, 0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, -1, 0, 0, 0, 0, 3]
    ],
    "raw_congruence_basis": [
      [-24, -18, -24, -19, -23],
      [24, 12, 0, 17, 12],
      [0, 6, 0, 1, 5],
      [0, 0, 24, 0, 5],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
    "predicted_basis": [
      [-24, -18, -24, -19, -23],
      [24, 12, 0, 17, 12],
      [0, 6, 0, 1, 5],
      [0, 0, 24, 0, 5],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
```

```
    "full_norm_image": [
      [-96, -82, -32, -73, -12, -105, -100, -40],
      [96, 80, 0, 71, 0, 80, 92, 0],
      [24, 22, 8, 19, 12, 27, 28, 16],
      [0, 0, 32, 0, 0, 23, 0, 28],
      [-24, -20, 0, -17, 0, -20, -20, 0],
      [0, 0, -8, 0, 0, -5, 0, -4]
    ],
    "common_deleted_row_HNF": [
      [24, 0, 6, 5, 5],
      [0, 24, 12, 12, 17],
      [0, 0, 6, 1, 1],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
    "image_preimage_matrix_C": [
      [1, -2, 0, 0, -2],
      [-8, -1, -8, -6, -4],
      [0, 0, -5, 0, -2],
      [8, 4, 0, 7, 4],
      [0, 0, 0, 0, 0],
      [0, 0, 8, 0, 3],
      [0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0]
    ],
    "predicted_coordinates_of_image_D": [
      [17, 14, 0, 12, -1, 14, 14, -1],
      [8, 7, 8, 6, 2, 12, 8, 6],
      [0, 0, 3, 0, 0, 2, 0, 2],
      [-24, -20, 0, -17, 0, -20, -20, 0],
      [0, 0, -8, 0, 0, -5, 0, -4]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 2,
    "tame": 5,
    "lower_divisors": [1, 3, 5, 15],
    "upper_divisors": [1, 3, 5, 9, 15, 45],
    "lower_eta_basis": [[-12, -6, -13], [12, 0, 7], [0, 6, 5], [0, 0, 1]],
    "upper_eta_basis": [
      [-12, -10, -3, -5, -11],
      [12, 8, 0, 3, 8],
      [0, 2, 0, 1, 1],
      [0, 0, 3, 0, 1],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
    "norm_matrix": [[4, 3, 0, 0, 0, 0], [-1, 0, 0, 3, 0, 0], [0, 0, 4, 0, 3, 0], [0, 0, -1,
0, 0, 3]],
    "raw_congruence_basis": [[-12, -6, -13], [12, 0, 7], [0, 6, 5], [0, 0, 1]],
    "predicted_basis": [[-12, -6, -13], [12, 0, 7], [0, 6, 5], [0, 0, 1]],
    "full_norm_image": [
      [-12, -16, -12, -11, -20],
      [12, 10, 12, 5, 14],
      [0, 8, 0, 7, 4],
      [0, -2, 0, -1, 2]
```

```
    ],
    "common_deleted_row_HNF": [[12, 0, 7], [0, 6, 5], [0, 0, 1]],
    "image_preimage_matrix_C": [[1, 0, 1], [0, -1, -2], [0, 0, 0], [0, 2, 3], [0, 0, 0]],
    "predicted_coordinates_of_image_D": [[1, 2, 1, 1, 0], [0, 3, 0, 2, -1], [0, -2, 0, -1,
2]],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 2,
    "tame": 10,
    "lower_divisors": [1, 2, 3, 5, 6, 10, 15, 30],
    "upper_divisors": [1, 2, 3, 5, 6, 9, 10, 15, 18, 30, 45, 90],
    "lower_eta_basis": [
      [-24, -12, -18, -13, -13, -25, -30],
      [24, 0, 16, 5, 11, 16, 17],
      [0, 12, 0, 7, 0, 7, 11],
      [0, 0, 2, 0, 1, 1, 1],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-24, -20, -18, -17, -3, -13, -5, -25, -14, -19, -15],
      [24, 16, 16, 13, 0, 11, 0, 23, 9, 16, 11],
      [0, 4, 0, 3, 0, 0, 3, 0, 3, 0, 0],
      [0, 0, 2, 0, 0, 1, 1, 0, 1, 1, 1],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0],
      [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
      [0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0],
      [0, -1, 0, 0, 0, 0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 0, 0, 4, 0, 0, 3, 0],
      [0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 3],
      [0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0]
    ],
    "raw_congruence_basis": [
      [-24, -12, -6, -13, -15, -13, -32],
      [24, 0, 0, 5, 9, 0, 15],
      [0, 12, 0, 7, 0, 7, 11],
      [0, 0, 6, 0, 3, 5, 3],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 2],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "predicted_basis": [
```

```
      [-24, -12, -6, -13, -15, -13, -32],
      [24, 0, 0, 5, 9, 0, 15],
      [0, 12, 0, 7, 0, 7, 11],
      [0, 0, 6, 0, 3, 5, 3],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 2],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "full_norm_image": [
      [-96, -68, -72, -59, -12, -52, -11, -100, -47, -76, -60],
      [96, 64, 64, 55, 0, 44, 0, 92, 36, 64, 44],
      [24, 20, 18, 17, 12, 13, 5, 28, 14, 22, 21],
      [0, 0, 8, 0, 0, 4, 7, 0, 4, 4, 4],
      [-24, -16, -16, -13, 0, -11, 0, -20, -9, -16, -11],
      [0, 0, 0, 0, 0, 4, 0, 0, 3, 0, 0],
      [0, 0, -2, 0, 0, -1, -1, 0, -1, 2, -1],
      [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 3]
    ],
    "common_deleted_row_HNF": [
      [24, 0, 6, 0, 5, 5, 11],
      [0, 12, 6, 9, 11, 6, 3],
      [0, 0, 6, 3, 1, 1, 4],
      [0, 0, 0, 3, 0, 0, 2],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "image_preimage_matrix_C": [
      [1, -2, 2, -1, 2, 2, 1],
      [-8, 3, -2, -1, -5, -1, -5],
      [0, 0, -1, 0, -1, -2, -1],
      [8, 0, 0, 3, 3, 0, 5],
      [0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, -1],
      [0, 0, 2, 0, 1, 3, 1],
      [0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 2],
      [0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0]
    ],
    "predicted_coordinates_of_image_D": [
      [9, 6, 6, 5, 0, 4, 0, 8, 3, 6, 3],
      [16, 11, 12, 9, 1, 9, 1, 14, 7, 10, 6],
      [0, 0, 3, 0, 0, 1, 2, 0, 1, -1, 1],
      [-24, -16, -16, -13, 0, -11, 0, -20, -9, -16, -11],
      [0, 0, 0, 0, 0, 2, 0, 0, 1, 0, -2],
      [0, 0, -2, 0, 0, -1, -1, 0, -1, 2, -1],
      [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 3]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 3,
    "tame": 1,
    "lower_divisors": [1, 3, 9],
    "upper_divisors": [1, 3, 9, 27],
```

```
  "lower_eta_basis": [[-12, -3], [12, 0], [0, 3]],
  "upper_eta_basis": [[-12, -9, -12], [12, 8, 9], [0, 1, 0], [0, 0, 3]],
  "norm_matrix": [[4, 3, 0, 0], [-1, 0, 3, 0], [0, 0, 0, 3]],
  "raw_congruence_basis": [[-12, -3], [12, 0], [0, 3]],
  "predicted_basis": [[-12, -9], [12, 0], [0, 9]],
  "full_norm_image": [[-12, -12, -21], [12, 12, 12], [0, 0, 9]],
  "common_deleted_row_HNF": [[9, 6], [0, 12]],
  "image_preimage_matrix_C": [[1, -1], [0, 0], [0, 1]],
  "predicted_coordinates_of_image_D": [[1, 1, 1], [0, 0, 1]],
  "image_times_C_equals_predicted": true,
  "predicted_times_D_equals_image": true,
  "all_equal": true
},
{
  "depth": 3,
  "tame": 2,
  "lower_divisors": [1, 2, 3, 6, 9, 18],
  "upper_divisors": [1, 2, 3, 6, 9, 18, 27, 54],
  "lower_eta_basis": [
    [-24, -20, -17, -3, -25],
    [24, 16, 13, 0, 23],
    [0, 4, 3, 0, 0],
    [0, 0, 1, 0, 0],
    [0, 0, 0, 3, 1],
    [0, 0, 0, 0, 1]
  ],
  "upper_eta_basis": [
    [-24, -20, -17, -17, -8, -20, -17],
    [24, 16, 13, 16, 7, 16, 13],
    [0, 4, 3, 0, 0, 1, 2],
    [0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 3, 1],
    [0, 0, 0, 0, 0, 0, 1]
  ],
  "norm_matrix": [
    [4, 0, 3, 0, 0, 0, 0, 0],
    [0, 4, 0, 3, 0, 0, 0, 0],
    [-1, 0, 0, 0, 3, 0, 0, 0],
    [0, -1, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "raw_congruence_basis": [
    [-24, -12, -13, -3, -24],
    [24, 0, 5, 0, 21],
    [0, 12, 7, 0, 0],
    [0, 0, 1, 0, 0],
    [0, 0, 0, 3, 0],
    [0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
    [-24, -12, -13, -9, -27],
    [24, 0, 5, 0, 21],
    [0, 12, 7, 0, 0],
    [0, 0, 1, 0, 0],
    [0, 0, 0, 9, 3],
    [0, 0, 0, 0, 3]
```

```
    ],
    "full_norm_image": [
      [-96, -68, -59, -68, -32, -77, -62],
      [96, 64, 55, 64, 28, 64, 52],
      [24, 20, 17, 20, 8, 20, 17],
      [-24, -16, -13, -16, -4, -16, -13],
      [0, 0, 0, 0, 0, 9, 3],
      [0, 0, 0, 0, 0, 0, 3]
    ],
    "common_deleted_row_HNF": [
      [72, 24, 60, 35, 21],
      [0, 3, 0, 2, 0],
      [0, 0, 12, 7, 0],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 3]
    ],
    "image_preimage_matrix_C": [
      [1, -2, -1, 0, 1],
      [-8, 3, -1, -1, -8],
      [8, 0, 3, 0, 7],
      [0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [9, 6, 5, 6, 2, 6, 4],
      [16, 11, 9, 11, 3, 11, 9],
      [-24, -16, -13, -16, -4, -16, -13],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 3,
    "tame": 4,
    "lower_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 36],
    "upper_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, 108],
    "lower_eta_basis": [
      [-24, -22, -8, -19, -3, -27, -25, -10],
      [24, 20, 0, 17, 0, 20, 23, 0],
      [0, 2, 0, 1, 0, 1, 0, 0],
      [0, 0, 8, 0, 0, 5, 0, 7],
      [0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 1, 2],
      [0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-24, -22, -8, -19, -17, -27, -8, -20, -24, -19, -25],
      [24, 20, 0, 17, 16, 20, 7, 16, 16, 17, 16],
      [0, 2, 0, 1, 0, 1, 0, 1, 0, 0, 1],
      [0, 0, 8, 0, 0, 5, 0, 0, 7, 0, 5],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
```

```
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "norm_matrix": [
    [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 4, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, -1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, -1, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "raw_congruence_basis": [
    [-24, -18, -24, -19, -3, -23, -24, -24],
    [24, 12, 0, 17, 0, 12, 21, 0],
    [0, 6, 0, 1, 0, 5, 0, 0],
    [0, 0, 24, 0, 0, 5, 0, 21],
    [0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
    [-24, -18, -24, -19, -9, -23, -27, -30],
    [24, 12, 0, 17, 0, 12, 21, 0],
    [0, 6, 0, 1, 0, 5, 0, 0],
    [0, 0, 24, 0, 0, 5, 0, 21],
    [0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 9, 0, 3, 6],
    [0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [-96, -82, -32, -73, -68, -105, -32, -77, -96, -76, -97],
    [96, 80, 0, 71, 64, 80, 28, 64, 64, 68, 64],
    [24, 22, 8, 19, 20, 27, 8, 20, 24, 19, 25],
    [0, 0, 32, 0, 0, 23, 0, 0, 28, 0, 20],
    [-24, -20, 0, -17, -16, -20, -4, -16, -16, -17, -16],
    [0, 0, 0, 0, 0, 0, 0, 9, 0, 3, 6],
    [0, 0, -8, 0, 0, -5, 0, 0, -4, 0, -5],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 48, 54, 24, 53, 45, 25, 0],
    [0, 24, 12, 0, 17, 0, 12, 21],
    [0, 0, 6, 0, 1, 0, 5, 0],
    [0, 0, 0, 3, 0, 0, 2, 0],
    [0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0],
```

```
      [0, 0, 0, 0, 0, 0, 0, 3]
    ],
    "image_preimage_matrix_C": [
      [1, -2, 0, 0, 1, -2, 1, 1],
      [-8, -1, -8, -6, -2, -4, -8, -9],
      [0, 0, -5, 0, 0, -2, 0, -5],
      [8, 4, 0, 7, 0, 4, 7, 0],
      [0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 8, 0, 0, 3, 0, 7],
      [0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [17, 14, 0, 12, 11, 14, 3, 11, 11, 11, 11],
      [8, 7, 8, 6, 6, 12, 2, 6, 10, 6, 11],
      [0, 0, 3, 0, 0, 2, 0, 0, 2, 0, 1],
      [-24, -20, 0, -17, -16, -20, -4, -16, -16, -17, -16],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, -8, 0, 0, -5, 0, 0, -4, 0, -5],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 3,
    "tame": 5,
    "lower_divisors": [1, 3, 5, 9, 15, 45],
    "upper_divisors": [1, 3, 5, 9, 15, 27, 45, 135],
    "lower_eta_basis": [
      [-12, -10, -3, -5, -11],
      [12, 8, 0, 3, 8],
      [0, 2, 0, 1, 1],
      [0, 0, 3, 0, 1],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-12, -10, -9, -5, -12, -2, -5],
      [12, 8, 8, 3, 9, 0, 2],
      [0, 2, 0, 1, 0, 1, 1],
      [0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 1],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 3, 0, 0, 0, 0, 0, 0],
      [-1, 0, 0, 3, 0, 0, 0, 0],
      [0, 0, 4, 0, 3, 0, 0, 0],
      [0, 0, 0, 0, 0, 3, 0, 0],
      [0, 0, -1, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 0, 3]
```

```
    ],
    "raw_congruence_basis": [
      [-12, -6, -3, -13, -6],
      [12, 0, 0, 7, 0],
      [0, 6, 0, 5, 3],
      [0, 0, 3, 0, 0],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 3]
    ],
    "predicted_basis": [
      [-12, -6, -9, -13, -9],
      [12, 0, 0, 7, 0],
      [0, 6, 0, 5, 3],
      [0, 0, 9, 0, 3],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 3]
    ],
    "full_norm_image": [
      [-12, -16, -12, -11, -21, -8, -14],
      [12, 10, 12, 5, 12, 2, 5],
      [0, 8, 0, 7, 0, 4, 4],
      [0, 0, 0, 0, 9, 0, 3],
      [0, -2, 0, -1, 0, 2, -1],
      [0, 0, 0, 0, 0, 0, 3]
    ],
    "common_deleted_row_HNF": [
      [18, 6, 6, 3, 17],
      [0, 12, 0, 0, 7],
      [0, 0, 3, 0, 2],
      [0, 0, 0, 3, 0],
      [0, 0, 0, 0, 1]
    ],
    "image_preimage_matrix_C": [
      [1, 0, -1, 1, 0],
      [0, -1, 0, -2, -1],
      [0, 0, 0, 0, 0],
      [0, 2, 0, 3, 1],
      [0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [1, 2, 1, 1, 1, -1, 1],
      [0, 3, 0, 2, 0, -1, 1],
      [0, 0, 0, 0, 1, 0, 0],
      [0, -2, 0, -1, 0, 2, -1],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 3,
    "tame": 10,
    "lower_divisors": [1, 2, 3, 5, 6, 9, 10, 15, 18, 30, 45, 90],
    "upper_divisors": [1, 2, 3, 5, 6, 9, 10, 15, 18, 27, 30, 45, 54, 90, 135, 270],
    "lower_eta_basis": [
      [-24, -20, -18, -17, -3, -13, -5, -25, -14, -19, -15],
```

```
    [24, 16, 16, 13, 0, 11, 0, 23, 9, 16, 11],
    [0, 4, 0, 3, 0, 0, 3, 0, 3, 0, 0],
    [0, 0, 2, 0, 0, 1, 1, 0, 1, 1, 1],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "upper_eta_basis": [
    [-24, -20, -18, -17, -17, -13, -5, -8, -20, -14, -2, -17, -5, -5, -14],
    [24, 16, 16, 13, 16, 11, 0, 7, 16, 9, 0, 13, 3, 0, 9],
    [0, 4, 0, 3, 0, 0, 3, 0, 1, 3, 0, 2, 0, 2, 1],
    [0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "norm_matrix": [
    [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, -1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "raw_congruence_basis": [
    [-24, -12, -6, -13, -3, -15, -13, -24, -32, -6, -15],
    [24, 0, 0, 5, 0, 9, 0, 21, 15, 0, 9],
    [0, 12, 0, 7, 0, 0, 7, 0, 11, 0, 0],
    [0, 0, 6, 0, 0, 3, 5, 0, 3, 3, 3],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 2, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
```

```
    [-24, -12, -6, -13, -9, -15, -13, -27, -32, -9, -21],
    [24, 0, 0, 5, 0, 9, 0, 21, 15, 0, 9],
    [0, 12, 0, 7, 0, 0, 7, 0, 11, 0, 0],
    [0, 0, 6, 0, 0, 3, 5, 0, 3, 3, 3],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 9, 0, 0, 3, 0, 3, 6],
    [0, 0, 0, 0, 0, 3, 0, 0, 2, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [-96, -68, -72, -59, -68, -52, -11, -32, -77, -47, -8, -62, -20, -14, -53],
    [96, 64, 64, 55, 64, 44, 0, 28, 64, 36, 0, 52, 12, 0, 36],
    [24, 20, 18, 17, 20, 13, 5, 8, 20, 14, 2, 17, 5, 5, 14],
    [0, 0, 8, 0, 0, 4, 7, 0, 0, 4, 4, 0, 4, 4, 4],
    [-24, -16, -16, -13, -16, -11, 0, -4, -16, -9, 0, -13, -3, 0, -9],
    [0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 3, 0, 3, 6],
    [0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, -2, 0, 0, -1, -1, 0, 0, -1, 2, 0, -1, -1, -1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 48, 60, 57, 59, 45, 24, 50, 0, 31, 9],
    [0, 6, 0, 3, 5, 0, 0, 3, 3, 0, 3],
    [0, 0, 12, 0, 7, 0, 0, 7, 0, 11, 0],
    [0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 2, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, -2, 2, -1, 0, 2, 2, 1, 1, 2, 2],
    [-8, 3, -2, -1, -1, -5, -1, -8, -5, -2, -5],
    [0, 0, -1, 0, 0, -1, -2, 0, -1, -1, -1],
    [8, 0, 0, 3, 0, 3, 0, 7, 5, 0, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0],
    [0, 0, 2, 0, 0, 1, 3, 0, 1, 1, 1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "predicted_coordinates_of_image_D": [
    [9, 6, 6, 5, 6, 4, 0, 2, 6, 3, 0, 4, 0, 0, 3],
```

```
      [16, 11, 12, 9, 11, 9, 1, 3, 11, 7, -1, 9, 0, 1, 7],
      [0, 0, 3, 0, 0, 1, 2, 0, 0, 1, -1, 0, 1, 1, 1],
      [-24, -16, -16, -13, -16, -11, 0, -4, -16, -9, 0, -13, -3, 0, -9],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, -2, 0, 0],
      [0, 0, -2, 0, 0, -1, -1, 0, 0, -1, 2, 0, -1, -1, -1],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 4,
    "tame": 1,
    "lower_divisors": [1, 3, 9, 27],
    "upper_divisors": [1, 3, 9, 27, 81],
    "lower_eta_basis": [[-12, -9, -12], [12, 8, 9], [0, 1, 0], [0, 0, 3]],
    "upper_eta_basis": [[-12, -9, -12, -3], [12, 8, 11, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0,
0, 0, 3]],
    "norm_matrix": [[4, 3, 0, 0, 0], [-1, 0, 3, 0, 0], [0, 0, 0, 3, 0], [0, 0, 0, 0, 3]],
    "raw_congruence_basis": [[-12, -3, -12], [12, 0, 9], [0, 3, 0], [0, 0, 3]],
    "predicted_basis": [[-12, -3, -12], [12, 0, 3], [0, 3, 0], [0, 0, 9]],
    "full_norm_image": [[-12, -12, -15, -12], [12, 12, 12, 3], [0, 0, 3, 0], [0, 0, 0, 9]],
    "common_deleted_row_HNF": [[36, 24, 33], [0, 3, 0], [0, 0, 3]],
    "image_preimage_matrix_C": [[1, -1, 0], [0, 0, 0], [0, 1, 0], [0, 0, 1]],
    "predicted_coordinates_of_image_D": [[1, 1, 1, 0], [0, 0, 1, 0], [0, 0, 0, 1]],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 4,
    "tame": 2,
    "lower_divisors": [1, 2, 3, 6, 9, 18, 27, 54],
    "upper_divisors": [1, 2, 3, 6, 9, 18, 27, 54, 81, 162],
    "lower_eta_basis": [
      [-24, -20, -17, -17, -8, -20, -17],
      [24, 16, 13, 16, 7, 16, 13],
      [0, 4, 3, 0, 0, 1, 2],
      [0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 3, 1],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-24, -20, -17, -17, -8, -20, -17, -3, -25],
      [24, 16, 13, 16, 7, 16, 13, 0, 23],
      [0, 4, 3, 0, 0, 3, 3, 0, 0],
      [0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 3, 1],
```

```
    [0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "norm_matrix": [
    [4, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, 4, 0, 3, 0, 0, 0, 0, 0, 0],
    [-1, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, -1, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "raw_congruence_basis": [
    [-24, -12, -13, -3, -24, -12, -27],
    [24, 0, 5, 0, 21, 0, 15],
    [0, 12, 7, 0, 0, 9, 9],
    [0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
    [-24, -12, -13, -3, -24, -12, -27],
    [24, 0, 5, 0, 21, 0, 15],
    [0, 12, 7, 0, 0, 3, 6],
    [0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 9, 3],
    [0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [-96, -68, -59, -68, -32, -71, -59, -12, -100],
    [96, 64, 55, 64, 28, 64, 52, 0, 92],
    [24, 20, 17, 20, 8, 20, 17, 3, 25],
    [-24, -16, -13, -16, -4, -16, -13, 0, -23],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 9, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 24, 15, 5, 69, 24, 39],
    [0, 12, 9, 11, 0, 9, 9],
    [0, 0, 3, 1, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, -2, -1, 0, 1, 0, -1],
    [-8, 3, -1, -1, -8, 0, -4],
    [8, 0, 3, 0, 7, 0, 5],
    [0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0],
```

```
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [9, 6, 5, 6, 2, 6, 4, 0, 8],
      [16, 11, 9, 11, 3, 11, 9, 0, 15],
      [-24, -16, -13, -16, -4, -16, -13, 0, -23],
      [0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 4,
    "tame": 4,
    "lower_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, 108],
    "upper_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, 81, 108, 162, 324],
    "lower_eta_basis": [
      [-24, -22, -8, -19, -17, -27, -8, -20, -24, -19, -25],
      [24, 20, 0, 17, 16, 20, 7, 16, 16, 17, 16],
      [0, 2, 0, 1, 0, 1, 0, 1, 0, 0, 1],
      [0, 0, 8, 0, 0, 5, 0, 0, 7, 0, 5],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-24, -22, -8, -19, -17, -27, -8, -22, -24, -19, -3, -27, -25, -10],
      [24, 20, 0, 17, 16, 20, 7, 20, 16, 17, 0, 20, 23, 0],
      [0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0],
      [0, 0, 8, 0, 0, 5, 0, 0, 7, 0, 0, 5, 0, 7],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, -1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
```

```
    [0, 0, 0, -1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "raw_congruence_basis": [
    [-24, -18, -24, -19, -3, -23, -24, -18, -24, -9, -33],
    [24, 12, 0, 17, 0, 12, 21, 12, 0, 3, 12],
    [0, 6, 0, 1, 0, 5, 0, 3, 0, 3, 3],
    [0, 0, 24, 0, 0, 5, 0, 0, 21, 0, 15],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
    [-24, -18, -24, -19, -3, -23, -24, -12, -24, -9, -27],
    [24, 12, 0, 17, 0, 12, 21, 0, 0, 3, 0],
    [0, 6, 0, 1, 0, 5, 0, 3, 0, 0, 3],
    [0, 0, 24, 0, 0, 5, 0, 0, 21, 0, 15],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 9, 0, 3, 6],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [-96, -82, -32, -73, -68, -105, -32, -85, -96, -73, -12, -105, -100, -40],
    [96, 80, 0, 71, 64, 80, 28, 80, 64, 68, 0, 80, 92, 0],
    [24, 22, 8, 19, 20, 27, 8, 22, 24, 19, 3, 27, 25, 10],
    [0, 0, 32, 0, 0, 23, 0, 0, 28, 0, 0, 20, 0, 28],
    [-24, -20, 0, -17, -16, -20, -4, -20, -16, -17, 0, -20, -23, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, -8, 0, 0, -5, 0, 0, -4, 0, 0, -5, 0, -7],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 3, 6],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 48, 54, 63, 53, 69, 58, 48, 30, 63, 15],
    [0, 24, 12, 12, 17, 0, 0, 21, 12, 12, 3],
    [0, 0, 6, 3, 1, 0, 2, 0, 3, 3, 3],
    [0, 0, 0, 3, 0, 0, 2, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
```

```
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
    ],
    "image_preimage_matrix_C": [
      [1, -2, 0, 0, 0, -2, 1, 0, 0, 0, 0],
      [-8, -1, -8, -6, -1, -4, -8, 0, -8, -2, -5],
      [0, 0, -5, 0, 0, -2, 0, 0, -5, 0, -4],
      [8, 4, 0, 7, 0, 4, 7, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 8, 0, 0, 3, 0, 0, 7, 0, 5],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [17, 14, 0, 12, 11, 14, 3, 14, 11, 11, 0, 14, 16, 0],
      [8, 7, 8, 6, 6, 12, 2, 7, 10, 6, 0, 12, 8, 7],
      [0, 0, 3, 0, 0, 2, 0, 0, 2, 0, 0, 1, 0, 2],
      [-24, -20, 0, -17, -16, -20, -4, -20, -16, -17, 0, -20, -23, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, -8, 0, 0, -5, 0, 0, -4, 0, 0, -5, 0, -7],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 4,
    "tame": 5,
    "lower_divisors": [1, 3, 5, 9, 15, 27, 45, 135],
    "upper_divisors": [1, 3, 5, 9, 15, 27, 45, 81, 135, 405],
    "lower_eta_basis": [
      [-12, -10, -9, -5, -12, -2, -5],
      [12, 8, 8, 3, 9, 0, 2],
      [0, 2, 0, 1, 0, 1, 1],
      [0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 1],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-12, -10, -9, -5, -12, -2, -3, -5, -11],
      [12, 8, 8, 3, 11, 0, 0, 3, 8],
      [0, 2, 0, 1, 0, 1, 0, 1, 1],
      [0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0],
```

```
      [0, 0, 0, 0, 0, 0, 3, 0, 1],
      [0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 3, 0, 0, 0, 0, 0, 0, 0, 0],
      [-1, 0, 0, 3, 0, 0, 0, 0, 0, 0],
      [0, 0, 4, 0, 3, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
      [0, 0, -1, 0, 0, 0, 3, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
    ],
    "raw_congruence_basis": [
      [-12, -6, -3, -13, -12, -6, -15],
      [12, 0, 0, 7, 9, 0, 9],
      [0, 6, 0, 5, 0, 3, 3],
      [0, 0, 3, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 3]
    ],
    "predicted_basis": [
      [-12, -6, -3, -13, -12, -6, -15],
      [12, 0, 0, 7, 3, 0, 6],
      [0, 6, 0, 5, 0, 3, 3],
      [0, 0, 3, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 9, 0, 3],
      [0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 3]
    ],
    "full_norm_image": [
      [-12, -16, -12, -11, -15, -8, -12, -11, -20],
      [12, 10, 12, 5, 12, 2, 3, 5, 11],
      [0, 8, 0, 7, 0, 4, 0, 4, 4],
      [0, 0, 0, 0, 3, 0, 0, 0, 0],
      [0, -2, 0, -1, 0, 2, 0, -1, -1],
      [0, 0, 0, 0, 0, 0, 9, 0, 3],
      [0, 0, 0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3]
    ],
    "common_deleted_row_HNF": [
      [36, 12, 21, 33, 2, 0, 21],
      [0, 6, 3, 0, 4, 3, 3],
      [0, 0, 3, 0, 2, 0, 0],
      [0, 0, 0, 3, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 3]
    ],
    "image_preimage_matrix_C": [
      [1, 0, -1, 1, 0, 0, 0],
      [0, -1, 0, -2, 0, -1, -1],
      [0, 0, 0, 0, 0, 0, 0],
      [0, 2, 0, 3, 0, 1, 1],
      [0, 0, 1, 0, 0, 0, 0],
```

```
      [0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [1, 2, 1, 1, 1, -1, 0, 1, 1],
      [0, 3, 0, 2, 0, -1, 0, 1, 1],
      [0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, -2, 0, -1, 0, 2, 0, -1, -1],
      [0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 4,
    "tame": 10,
    "lower_divisors": [1, 2, 3, 5, 6, 9, 10, 15, 18, 27, 30, 45, 54, 90, 135, 270],
    "upper_divisors": [1, 2, 3, 5, 6, 9, 10, 15, 18, 27, 30, 45, 54, 81, 90, 135, 162, 270,
405, 810],
    "lower_eta_basis": [
      [-24, -20, -18, -17, -17, -13, -5, -8, -20, -14, -2, -17, -5, -5, -14],
      [24, 16, 16, 13, 16, 11, 0, 7, 16, 9, 0, 13, 3, 0, 9],
      [0, 4, 0, 3, 0, 0, 3, 0, 1, 3, 0, 2, 0, 2, 1],
      [0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [
        -24, -20, -18, -17, -17, -13, -5, -8, -20, -14, -2, -17, -3, -5, -5, -25, -14,
        -19, -15
      ],
      [24, 16, 16, 13, 16, 11, 0, 7, 16, 9, 0, 13, 0, 3, 0, 23, 9, 16, 11],
      [0, 4, 0, 3, 0, 0, 3, 0, 3, 3, 0, 3, 0, 0, 3, 0, 3, 0, 0],
      [0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
```

```
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
],
"norm_matrix": [
  [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, -1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"raw_congruence_basis": [
  [-24, -12, -6, -13, -3, -15, -13, -24, -12, -32, -6, -27, -15, -15, -18],
  [24, 0, 0, 5, 0, 9, 0, 21, 0, 15, 0, 15, 9, 0, 3],
  [0, 12, 0, 7, 0, 0, 7, 0, 9, 11, 0, 9, 0, 9, 9],
  [0, 0, 6, 0, 0, 3, 5, 0, 0, 3, 3, 0, 3, 3, 3],
  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"predicted_basis": [
  [-24, -12, -6, -13, -3, -15, -13, -24, -12, -32, -6, -27, -15, -15, -18],
  [24, 0, 0, 5, 0, 9, 0, 21, 0, 15, 0, 15, 9, 0, 3],
  [0, 12, 0, 7, 0, 0, 7, 0, 3, 11, 0, 6, 0, 6, 3],
  [0, 0, 6, 0, 0, 3, 5, 0, 0, 3, 3, 0, 3, 3, 3],
  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 3, 0, 3, 6],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
```

```
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [
      -96, -68, -72, -59, -68, -52, -11, -32, -71, -47, -8, -59, -12, -20, -11,
      -100, -47, -76, -60
    ],
    [96, 64, 64, 55, 64, 44, 0, 28, 64, 36, 0, 52, 0, 12, 0, 92, 36, 64, 44],
    [24, 20, 18, 17, 20, 13, 5, 8, 20, 14, 2, 17, 3, 5, 5, 25, 14, 19, 15],
    [0, 0, 8, 0, 0, 4, 7, 0, 0, 4, 4, 0, 0, 4, 4, 0, 4, 4, 4],
    [
      -24, -16, -16, -13, -16, -11, 0, -4, -16, -9, 0, -13, 0, -3, 0, -23, -9, -16,
      -11
    ],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, -2, 0, 0, -1, -1, 0, 0, -1, 2, 0, 0, -1, -1, 0, -1, -1, -1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 3, 0, 3, 6],
    [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 24, 6, 48, 5, 69, 15, 35, 24, 54, 58, 48, 63, 15, 3],
    [0, 12, 6, 9, 11, 0, 6, 3, 9, 6, 0, 9, 9, 6, 9],
    [0, 0, 6, 3, 1, 0, 3, 4, 0, 3, 2, 3, 3, 3, 0],
    [0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, -2, 2, -1, 0, 2, 2, 1, 0, 1, 2, -1, 2, 0, 1],
    [-8, 3, -2, -1, -1, -5, -1, -8, 0, -5, -2, -4, -5, 0, -2],
    [0, 0, -1, 0, 0, -1, -2, 0, 0, -1, -1, 0, -1, -1, -1],
    [8, 0, 0, 3, 0, 3, 0, 7, 0, 5, 0, 5, 3, 0, 1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0],
    [0, 0, 2, 0, 0, 1, 3, 0, 0, 1, 1, 0, 1, 1, 1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "predicted_coordinates_of_image_D": [
      [9, 6, 6, 5, 6, 4, 0, 2, 6, 3, 0, 4, 0, 0, 0, 8, 3, 6, 4],
      [16, 11, 12, 9, 11, 9, 1, 3, 11, 7, -1, 9, 0, 0, 1, 15, 7, 11, 8],
      [0, 0, 3, 0, 0, 1, 2, 0, 0, 1, -1, 0, 0, 1, 1, 0, 1, 1, 1],
      [
        -24, -16, -16, -13, -16, -11, 0, -4, -16, -9, 0, -13, 0, -3, 0, -23, -9, -16,
        -11
      ],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, -2, 0, 0, 0, 0, 0],
      [0, 0, -2, 0, 0, -1, -1, 0, 0, -1, 2, 0, 0, -1, -1, 0, -1, -1, -1],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 5,
    "tame": 1,
    "lower_divisors": [1, 3, 9, 27, 81],
    "upper_divisors": [1, 3, 9, 27, 81, 243],
    "lower_eta_basis": [[-12, -9, -12, -3], [12, 8, 11, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0,
0, 0, 3]],
    "upper_eta_basis": [
      [-12, -9, -12, -9, -12],
      [12, 8, 11, 8, 9],
      [0, 1, 0, 0, 0],
      [0, 0, 1, 0, 0],
      [0, 0, 0, 1, 0],
      [0, 0, 0, 0, 3]
    ],
    "norm_matrix": [
      [4, 3, 0, 0, 0, 0],
      [-1, 0, 3, 0, 0, 0],
      [0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 3]
    ],
    "raw_congruence_basis": [[-12, -3, -12, -3], [12, 0, 9, 0], [0, 3, 0, 0], [0, 0, 3, 0],
[0, 0, 0, 3]],
    "predicted_basis": [[-12, -3, -12, -9], [12, 0, 9, 0], [0, 3, 0, 0], [0, 0, 3, 0], [0,
0, 0, 9]],
    "full_norm_image": [
      [-12, -12, -15, -12, -21],
      [12, 12, 12, 9, 12],
```

```
      [0, 0, 3, 0, 0],
      [0, 0, 0, 3, 0],
      [0, 0, 0, 0, 9]
    ],
    "common_deleted_row_HNF": [[9, 6, 6, 6], [0, 12, 0, 9], [0, 0, 3, 0], [0, 0, 0, 3]],
    "image_preimage_matrix_C": [[1, -1, 0, -1], [0, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0],
[0, 0, 0, 1]],
    "predicted_coordinates_of_image_D": [[1, 1, 1, 0, 1], [0, 0, 1, 0, 0], [0, 0, 0, 1, 0],
[0, 0, 0, 0, 1]],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 5,
    "tame": 2,
    "lower_divisors": [1, 2, 3, 6, 9, 18, 27, 54, 81, 162],
    "upper_divisors": [1, 2, 3, 6, 9, 18, 27, 54, 81, 162, 243, 486],
    "lower_eta_basis": [
      [-24, -20, -17, -17, -8, -20, -17, -3, -25],
      [24, 16, 13, 16, 7, 16, 13, 0, 23],
      [0, 4, 3, 0, 0, 3, 3, 0, 0],
      [0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 3, 1],
      [0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-24, -20, -17, -17, -8, -20, -17, -17, -8, -20, -17],
      [24, 16, 13, 16, 7, 16, 13, 16, 7, 16, 13],
      [0, 4, 3, 0, 0, 3, 3, 0, 0, 1, 2],
      [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
      [-1, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
      [0, -1, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
    ],
    "raw_congruence_basis": [
      [-24, -12, -13, -3, -24, -12, -27, -3, -24],
      [24, 0, 5, 0, 21, 0, 15, 0, 21],
```

```
    [0, 12, 7, 0, 0, 9, 9, 0, 0],
    [0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
    [-24, -12, -13, -3, -24, -12, -27, -9, -27],
    [24, 0, 5, 0, 21, 0, 15, 0, 21],
    [0, 12, 7, 0, 0, 9, 9, 0, 0],
    [0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 9, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [-96, -68, -59, -68, -32, -71, -59, -68, -32, -77, -62],
    [96, 64, 55, 64, 28, 64, 52, 64, 28, 64, 52],
    [24, 20, 17, 20, 8, 20, 17, 17, 8, 20, 17],
    [-24, -16, -13, -16, -4, -16, -13, -16, -7, -16, -13],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 24, 60, 35, 69, 24, 60, 69, 21],
    [0, 3, 0, 2, 0, 0, 0, 0, 0],
    [0, 0, 12, 7, 0, 0, 9, 9, 0],
    [0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, -2, -1, 0, 1, -2, -1, 0, 1],
    [-8, 3, -1, -1, -8, 2, -3, -1, -8],
    [8, 0, 3, 0, 7, 0, 5, 0, 7],
    [0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "predicted_coordinates_of_image_D": [
    [9, 6, 5, 6, 2, 6, 4, 6, 2, 6, 4],
```

```
      [16, 11, 9, 11, 3, 11, 9, 10, 4, 11, 9],
      [-24, -16, -13, -16, -4, -16, -13, -16, -7, -16, -13],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 5,
    "tame": 4,
    "lower_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, 81, 108, 162, 324],
    "upper_divisors": [1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, 81, 108, 162, 243, 324, 486,
972],
    "lower_eta_basis": [
      [-24, -22, -8, -19, -17, -27, -8, -22, -24, -19, -3, -27, -25, -10],
      [24, 20, 0, 17, 16, 20, 7, 20, 16, 17, 0, 20, 23, 0],
      [0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0],
      [0, 0, 8, 0, 0, 5, 0, 0, 7, 0, 0, 5, 0, 7],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [
        -24, -22, -8, -19, -17, -27, -8, -22, -24, -19, -17, -27, -8, -20, -24, -19,
        -25
      ],
      [24, 20, 0, 17, 16, 20, 7, 20, 16, 17, 16, 20, 7, 16, 16, 17, 16],
      [0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1],
      [0, 0, 8, 0, 0, 5, 0, 0, 7, 0, 0, 5, 0, 0, 7, 0, 5],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
```

```
"norm_matrix": [
  [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, -1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, -1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"raw_congruence_basis": [
  [-24, -18, -24, -19, -3, -23, -24, -18, -24, -9, -3, -33, -24, -24],
  [24, 12, 0, 17, 0, 12, 21, 12, 0, 3, 0, 12, 21, 0],
  [0, 6, 0, 1, 0, 5, 0, 3, 0, 3, 0, 3, 0, 0],
  [0, 0, 24, 0, 0, 5, 0, 0, 21, 0, 0, 15, 0, 21],
  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"predicted_basis": [
  [-24, -18, -24, -19, -3, -23, -24, -18, -24, -9, -9, -33, -27, -30],
  [24, 12, 0, 17, 0, 12, 21, 12, 0, 3, 0, 12, 21, 0],
  [0, 6, 0, 1, 0, 5, 0, 3, 0, 3, 0, 3, 0, 0],
  [0, 0, 24, 0, 0, 5, 0, 0, 21, 0, 0, 15, 0, 21],
  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 3, 6],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"full_norm_image": [
  [
    -96, -82, -32, -73, -68, -105, -32, -85, -96, -73, -68, -105, -32, -77, -96,
    -76, -97
  ],
  [96, 80, 0, 71, 64, 80, 28, 80, 64, 68, 64, 80, 28, 64, 64, 68, 64],
  [24, 22, 8, 19, 20, 27, 8, 22, 24, 19, 17, 27, 8, 20, 24, 19, 25],
  [0, 0, 32, 0, 0, 23, 0, 0, 28, 0, 0, 20, 0, 0, 28, 0, 20],
```

```
    [
      -24, -20, 0, -17, -16, -20, -4, -20, -16, -17, -16, -20, -7, -16, -16, -17,
      -16
    ],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, -8, 0, 0, -5, 0, 0, -4, 0, 0, -5, 0, 0, -7, 0, -5],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 3, 6],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 48, 54, 24, 53, 69, 25, 48, 54, 24, 63, 45, 63, 0],
    [0, 24, 12, 0, 17, 0, 12, 21, 12, 0, 3, 0, 12, 21],
    [0, 0, 6, 0, 1, 0, 5, 0, 3, 0, 3, 0, 3, 0],
    [0, 0, 0, 3, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, -2, 0, 0, 0, -2, 1, -1, 0, -1, 1, -1, 1, 1],
    [-8, -1, -8, -6, -1, -4, -8, -3, -8, 0, -2, -8, -8, -9],
    [0, 0, -5, 0, 0, -2, 0, 0, -5, 0, 0, -4, 0, -5],
    [8, 4, 0, 7, 0, 4, 7, 4, 0, 1, 0, 4, 7, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 8, 0, 0, 3, 0, 0, 7, 0, 0, 5, 0, 7],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "predicted_coordinates_of_image_D": [
    [17, 14, 0, 12, 11, 14, 3, 14, 11, 11, 11, 14, 5, 11, 11, 11, 11],
    [8, 7, 8, 6, 6, 12, 2, 7, 10, 6, 5, 12, 2, 6, 12, 6, 11],
    [0, 0, 3, 0, 0, 2, 0, 0, 2, 0, 0, 1, 0, 0, 2, 0, 1],
    [
      -24, -20, 0, -17, -16, -20, -4, -20, -16, -17, -16, -20, -7, -16, -16, -17,
      -16
    ],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
      [0, 0, -8, 0, 0, -5, 0, 0, -4, 0, 0, -5, 0, 0, -7, 0, -5],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 5,
    "tame": 5,
    "lower_divisors": [1, 3, 5, 9, 15, 27, 45, 81, 135, 405],
    "upper_divisors": [1, 3, 5, 9, 15, 27, 45, 81, 135, 243, 405, 1215],
    "lower_eta_basis": [
      [-12, -10, -9, -5, -12, -2, -3, -5, -11],
      [12, 8, 8, 3, 11, 0, 0, 3, 8],
      [0, 2, 0, 1, 0, 1, 0, 1, 1],
      [0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 3, 0, 1],
      [0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [-12, -10, -9, -5, -12, -2, -9, -5, -12, -2, -5],
      [12, 8, 8, 3, 11, 0, 8, 3, 9, 0, 2],
      [0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 1],
      [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "norm_matrix": [
      [4, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [-1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 4, 0, 3, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
      [0, 0, -1, 0, 0, 0, 3, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
    ],
    "raw_congruence_basis": [
      [-12, -6, -3, -13, -12, -6, -3, -15, -6],
```

```
    [12, 0, 0, 7, 9, 0, 0, 9, 0],
    [0, 6, 0, 5, 0, 3, 0, 3, 3],
    [0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "predicted_basis": [
    [-12, -6, -3, -13, -12, -6, -9, -15, -9],
    [12, 0, 0, 7, 9, 0, 0, 9, 0],
    [0, 6, 0, 5, 0, 3, 0, 3, 3],
    [0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 9, 0, 3],
    [0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "full_norm_image": [
    [-12, -16, -12, -11, -15, -8, -12, -11, -21, -8, -14],
    [12, 10, 12, 5, 12, 2, 9, 5, 12, 2, 5],
    [0, 8, 0, 7, 0, 4, 0, 4, 0, 4, 4],
    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, -2, 0, -1, 0, 2, 0, -1, 0, -1, -1],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [18, 6, 6, 15, 17, 6, 6, 3, 15],
    [0, 12, 0, 0, 7, 9, 0, 0, 9],
    [0, 0, 3, 0, 2, 0, 0, 0, 0],
    [0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, 0, -1, 1, 0, 0, -1, 1, 0],
    [0, -1, 0, -2, 0, -1, 0, -1, -1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 2, 0, 3, 0, 1, 0, 1, 1],
    [0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "predicted_coordinates_of_image_D": [
```

```
      [1, 2, 1, 1, 1, -1, 0, 1, 1, 0, 1],
      [0, 3, 0, 2, 0, -1, 0, 1, 0, 1, 1],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
      [0, -2, 0, -1, 0, 2, 0, -1, 0, -1, -1],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "image_times_C_equals_predicted": true,
    "predicted_times_D_equals_image": true,
    "all_equal": true
  },
  {
    "depth": 5,
    "tame": 10,
    "lower_divisors": [1, 2, 3, 5, 6, 9, 10, 15, 18, 27, 30, 45, 54, 81, 90, 135, 162, 270,
405, 810],
    "upper_divisors": [
      1, 2, 3, 5, 6, 9, 10, 15, 18, 27, 30, 45, 54, 81, 90, 135, 162, 243, 270, 405,
      486, 810, 1215, 2430
    ],
    "lower_eta_basis": [
      [
        -24, -20, -18, -17, -17, -13, -5, -8, -20, -14, -2, -17, -3, -5, -5, -25, -14,
        -19, -15
      ],
      [24, 16, 16, 13, 16, 11, 0, 7, 16, 9, 0, 13, 0, 3, 0, 23, 9, 16, 11],
      [0, 4, 0, 3, 0, 0, 3, 0, 3, 3, 0, 3, 0, 0, 3, 0, 3, 0, 0],
      [0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
      [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
    ],
    "upper_eta_basis": [
      [
        -24, -20, -18, -17, -17, -13, -5, -8, -20, -14, -2, -17, -17, -5, -5, -8, -20,
        -14, -2, -17, -5, -5, -14
      ],
      [24, 16, 16, 13, 16, 11, 0, 7, 16, 9, 0, 13, 16, 3, 0, 7, 16, 9, 0, 13, 3, 0, 9],
      [0, 4, 0, 3, 0, 0, 3, 0, 3, 3, 0, 3, 0, 0, 3, 0, 1, 3, 0, 2, 0, 2, 1],
      [0, 0, 2, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 0, 1, 2],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "norm_matrix": [
    [4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 4, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [-1, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, -1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "raw_congruence_basis": [
    [
      -24, -12, -6, -13, -3, -15, -13, -24, -12, -32, -6, -27, -3, -15, -15, -24,
      -18, -6, -15
    ],
    [24, 0, 0, 5, 0, 9, 0, 21, 0, 15, 0, 15, 0, 9, 0, 21, 3, 0, 9],
    [0, 12, 0, 7, 0, 0, 7, 0, 9, 11, 0, 9, 0, 0, 9, 0, 9, 0, 0],
    [0, 0, 6, 0, 0, 3, 5, 0, 0, 3, 3, 0, 0, 3, 3, 0, 3, 3, 3],
    [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
```

```
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"predicted_basis": [
  [
    -24, -12, -6, -13, -3, -15, -13, -24, -12, -32, -6, -27, -9, -15, -15, -27,
    -18, -9, -21
  ],
  [24, 0, 0, 5, 0, 9, 0, 21, 0, 15, 0, 15, 0, 9, 0, 21, 3, 0, 9],
  [0, 12, 0, 7, 0, 0, 7, 0, 9, 11, 0, 9, 0, 0, 9, 0, 9, 0, 0],
  [0, 0, 6, 0, 0, 3, 5, 0, 0, 3, 3, 0, 0, 3, 3, 0, 3, 3, 3],
  [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 3, 0, 3, 6],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
],
"full_norm_image": [
  [
    -96, -68, -72, -59, -68, -52, -11, -32, -71, -47, -8, -59, -68, -20, -11, -32,
    -77, -47, -8, -62, -20, -14, -53
  ],
  [
    96, 64, 64, 55, 64, 44, 0, 28, 64, 36, 0, 52, 64, 12, 0, 28, 64, 36, 0, 52,
    12, 0, 36
  ],
  [
    24, 20, 18, 17, 20, 13, 5, 8, 20, 14, 2, 17, 17, 5, 5, 8, 20, 14, 2, 17, 5, 5,
    14
  ],
  [0, 0, 8, 0, 0, 4, 7, 0, 0, 4, 4, 0, 0, 4, 4, 0, 0, 4, 4, 0, 4, 4, 4],
  [
    -24, -16, -16, -13, -16, -11, 0, -4, -16, -9, 0, -13, -16, -3, 0, -7, -16, -9,
    0, -13, -3, 0, -9
  ],
  [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 4, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, -2, 0, 0, -1, -1, 0, 0, -1, 2, 0, 0, -1, -1, 0, 0, -1, -1, 0, -1, -1, -1],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 3, 0, 3, 6],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
```

```
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "common_deleted_row_HNF": [
    [72, 48, 60, 57, 59, 69, 24, 50, 48, 60, 31, 57, 21, 45, 24, 48, 0, 69, 9],
    [0, 6, 0, 3, 5, 0, 0, 3, 3, 0, 0, 3, 3, 0, 0, 3, 3, 0, 3],
    [0, 0, 12, 0, 7, 0, 0, 7, 0, 9, 11, 0, 9, 0, 0, 9, 0, 9, 0],
    [0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3]
  ],
  "image_preimage_matrix_C": [
    [1, -2, 2, -1, 0, 2, 2, 1, -2, 1, 2, -1, 0, 2, 0, 1, 0, 2, 2],
    [-8, 3, -2, -1, -1, -5, -1, -8, 2, -5, -2, -3, -1, -5, 1, -8, 0, -2, -5],
    [0, 0, -1, 0, 0, -1, -2, 0, 0, -1, -1, 0, 0, -1, -1, 0, -1, -1, -1],
    [8, 0, 0, 3, 0, 3, 0, 7, 0, 5, 0, 5, 0, 3, 0, 7, 1, 0, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 2, 0, 0, 1, 3, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
  ],
  "predicted_coordinates_of_image_D": [
    [9, 6, 6, 5, 6, 4, 0, 2, 6, 3, 0, 4, 6, 0, 0, 2, 6, 3, 0, 4, 1, 0, 3],
    [16, 11, 12, 9, 11, 9, 1, 3, 11, 7, -1, 9, 10, 0, 1, 4, 11, 7, 0, 9, 2, 1, 7],
    [0, 0, 3, 0, 0, 1, 2, 0, 0, 1, -1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1],
    [
      -24, -16, -16, -13, -16, -11, 0, -4, -16, -9, 0, -13, -16, -3, 0, -7, -16, -9,
      0, -13, -3, 0, -9
```

```
        ],
        [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, -2, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, -2, 0, 0, -1, -1, 0, 0, -1, 2, 0, 0, -1, -1, 0, 0, -1, -1, 0, -1, -1, -1],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
      ],
      "image_times_C_equals_predicted": true,
      "predicted_times_D_equals_image": true,
      "all_equal": true
    }
  ],
  "summary": {
    "lattice_cases": 20,
    "multiplier_pairs": 576,
    "status": "all exact assertions passed"
  }
}
```


References

[1] J. S. Balakrishnan and B. Mazur, with an appendix by N. Dogra, Ogg's torsion conjecture: Fifty years later, *Bull. Amer. Math. Soc.* (N.S.) **62** (2025), no. 2, 235–268. https://doi.org/10.1090/bull/1851.

[2] S. Bosch, W. Lütkebohmert, and M. Raynaud, *Néron Models*, Ergebnisse der Mathematik und ihrer Grenzgebiete (3), vol. 21, Springer, Berlin, 1990.

[3] B. Cais, Canonical integral structures on the de Rham cohomology of curves, *Ann. Inst. Fourier* **59** (2009), no. 6, 2255–2300. https://doi.org/10.5802/aif.2490.

[4] F. Calegari, V. Dimitrov, and Y. Tang, The unbounded denominators conjecture, *J. Amer. Math. Soc.* **38** (2025), 627–702. https://www.math.uchicago.edu/~fcale/papers/UDC.pdf.

[5] K. Česnavičius, The Manin constant in the semistable case, *Compos. Math.* **154** (2018), 1889–1920. https://doi.org/10.1112/S0010437X18007273.

[6] K. Česnavičius, A modular description of $X_0(n)$, *Algebra Number Theory* **11** (2017), no. 9, 2001–2089. https://doi.org/10.2140/ant.2017.11.2001.

[7] Z. Chi, S. Liu, and P. Nie, On the Manin constant, arXiv:2610.25672v2, 2026. https://arxiv.org/abs/2610.25672v2.

[8] P. L. Clark and X. Xarles, Local bounds for torsion points on abelian varieties, *Canad. J. Math.* **60** (2008), no. 3, 532–555. https://doi.org/10.4153/CJM-2008-026-x.

[9] B. Conrad, *Algebraic theory of q-expansions*, Math 248B course notes, Stanford University. https://math.stanford.edu/~conrad/248BPage/handouts/qexp.pdf.

[10] B. Conrad, *Semistable reduction for abelian varieties*, course notes, Stanford University. https://math.stanford.edu/~conrad/DarmonCM/2011Notes/SemistableReduction.pdf.

[11] B. Conrad and W. A. Stein, Component groups of purely toric quotients, *Math. Res. Lett.* **8** (2001), 745–766. The section and lemma numbers cited here follow the author version of April 21, 2003: https://wstein.org/papers/compgrp/compgrp.pdf.

[12] J. A. Csirik, *The kernel of the Eisenstein ideal*, Ph.D. thesis, University of California, Berkeley, 1999. http://www.csirik.net/csirik-thesis.pdf.

[13] S. Dasgupta and J. Voight, Heegner points and Sylvester's conjecture, in *Arithmetic Geometry*, Clay Mathematics Proceedings, vol. 8, American Mathematical Society, Providence, RI, 2009, 91–102. https://services.math.duke.edu/~dasgupta/papers/clay.pdf.

[14] P. Deligne and M. Rapoport, Les schémas de modules de courbes elliptiques, in *Modular Functions of One Variable II*, Lecture Notes in Mathematics, vol. 349, Springer, Berlin, 1973, 143–316.

[15] V. G. Drinfeld, Two theorems on modular curves, *Funct. Anal. Appl.* **7** (1973), no. 2, 155–156.

[16] B. Edixhoven, Comparison of integral structures on spaces of modular forms of weight two, and computation of spaces of forms mod 2 of weight one, with appendices by J.-F. Mestre and G. Wiese, *J. Inst. Math. Jussieu* **5** (2006), no. 1, 1–34. https://arxiv.org/abs/math/0312019.

[17] B. Edixhoven, On the Manin constants of modular elliptic curves, in *Arithmetic Algebraic Geometry (Texel, 1989)*, Progress in Mathematics, vol. 89, Birkhäuser, Boston, 1991, 25–39.

[18] B. Edixhoven, Minimal resolution and stable reduction of $X_0(N)$, *Ann. Inst. Fourier* **40** (1990), no. 1, 31–67. https://www.numdam.org/item/AIF_1990__40_1_31_0/.

[19] B. Edixhoven, Néron models and tame ramification, *Compositio Math.* **81** (1992), 291–306. https://www.numdam.org/item/CM_1992__81_3_291_0/.

[20] B. Edixhoven and P. Parent, Semistable reduction of modular curves associated with maximal subgroups in prime level, *Doc. Math.* **26** (2021), 231–269. https://doi.org/10.4171/DM/814.

[21] A. Grothendieck, M. Raynaud, and D. S. Rim, *Groupes de monodromie en géométrie algébrique, I (SGA 7 I)*, Lecture Notes in Mathematics, vol. 288, Springer, Berlin, 1972; especially Exposé IX, Modèles de Néron et monodromie.

[22] J.-W. Guo, Y. Yang, H. Yoo, and M. Yu, The rational cuspidal subgroup of $J_0(p^2M)$ with M squarefree, *Math. Nachr.* **296** (2023), no. 10, 4634–4655. https://doi.org/10.1002/mana.202200177.

[23] B. W. Jordan, K. A. Ribet, and A. J. Scholl, Modular curves and Néron models of generalized Jacobians, *Compos. Math.* **160** (2024), 945–981. https://doi.org/10.1112/S0010437X23007662.

[24] N. M. Katz, Higher congruences between modular forms, *Ann. of Math.* (2) **101** (1975), 332–367. https://doi.org/10.2307/1970994.

[25] N. M. Katz and B. Mazur, *Arithmetic Moduli of Elliptic Curves*, Annals of Mathematics Studies, vol. 108, Princeton University Press, Princeton, NJ, 1985.

[26] A. Kumar and M. Kuwata, Elliptic K3 surfaces associated with the product of two elliptic curves: Mordell–Weil lattices and their fields of definition, *Nagoya Math. J.* **228** (2017), 124–185. https://doi.org/10.1017/nmj.2016.56.

[27] S. Ling, On the $\mathbb{Q}$-rational cuspidal subgroup and the component group of $J_0(p^r)$, *Israel J. Math.* **99** (1997), 29–54. https://doi.org/10.1007/BF02760675.

[28] S. Ling and J. Oesterlé, The Shimura subgroup of $J_0(N)$, *Astérisque* **196–197** (1991), 171–203.

[29] D. J. Lorenzini, Torsion points on the modular Jacobian $J_0(N)$, *Compositio Math.* **96** (1995), no. 2, 149–172. https://www.numdam.org/item/CM_1995__96_2_149_0/.

[30] E. Lupoian, Two-torsion subgroups of some modular Jacobians, *Int. J. Number Theory* **20** (2024), no. 10, 2543–2573. https://doi.org/10.1142/S1793042124501227. Corrected author version: arXiv:2205.13017v3 (2025).

[31] H. Maass, *Lectures on Modular Functions of One Complex Variable*, notes by S. Lal, Tata Institute of Fundamental Research Lectures on Mathematics, vol. 29, Bombay, 1964. https://mathweb.tifr.res.in/Documents/Publications/Lectures/tifr29.pdf.

[32] R. S. Maier, On rationally parametrized modular equations, *J. Ramanujan Math. Soc.* **24** (2009), no. 1, 1–73. https://arxiv.org/abs/math/0611041.

[33] Yu. I. Manin, Parabolic points and zeta-functions of modular curves, *Math. USSR-Izv.* **6** (1972), no. 1, 19–64.

[34] B. Mazur, Modular curves and the Eisenstein ideal, *Inst. Hautes Études Sci. Publ. Math.* **47** (1977), 33–186. https://doi.org/10.1007/BF02684339.

[35] F. Najman, The rational torsion of $J_1(p)$ is generated by differences of rational cusps, Hexagon 2611.00991v1, 2026. https://hexagonmath.org/2611.00991v1.

[36] A. P. Ogg, Rational points on certain elliptic modular curves, in *Analytic Number Theory* (St. Louis, MO, 1972), Proc. Sympos. Pure Math., vol. 24, American Mathematical Society, Providence, RI, 1973, 221–231.

[37] A. P. Ogg, Diophantine equations and modular forms, *Bull. Amer. Math. Soc.* **81** (1975), 14–27. https://doi.org/10.1090/S0002-9904-1975-13623-8.

[38] M. Ohta, Eisenstein ideals and the rational torsion subgroups of modular Jacobian varieties II, *Tokyo J. Math.* **37** (2014), no. 2, 273–318. https://doi.org/10.3836/tjm/1422452795.

[39] M. Raynaud, Spécialisation du foncteur de Picard, *Inst. Hautes Études Sci. Publ. Math.* **38** (1970), 27–76. https://www.numdam.org/item/PMIHES_1970__38__27_0/.

[40] M. Raynaud, Hauteurs et isogénies, *Astérisque* **127** (1985), 199–234. https://www.numdam.org/item/AST_1985__127__199_0/.

[41] Y. Ren, Rational torsion subgroups of modular Jacobian varieties, *J. Number Theory* **190** (2018), 169–186. https://doi.org/10.1016/j.jnt.2018.02.009.

[42] K. A. Ribet, Raising the levels of modular representations, in *Séminaire de Théorie des Nombres, Paris 1987–88*, Progress in Mathematics, vol. 81, Birkhäuser, Boston, 1990, 259–271. https://math.berkeley.edu/~ribet/Articles/dpp.pdf.

[43] K. A. Ribet, On modular representations of $\mathrm{Gal}(\overline{\mathbf{Q}}/\mathbf{Q})$ arising from modular forms, *Invent. Math.* **100** (1990), 431–476. https://math.berkeley.edu/~ribet/Articles/invent_100.pdf.

[44] K. A. Ribet and W. A. Stein, *Lectures on Modular Forms and Hecke Operators*, course notes. https://wstein.org/edu/2011/581g/ribet-stein/main.pdf.

[45] K. A. Ribet and P. Wake, Another look at rational torsion of modular Jacobians, *Proc. Natl. Acad. Sci. USA* **119** (2022), no. 41, e2210032119. https://doi.org/10.1073/pnas.2210032119.

[46] J. Rouse and J. J. Webb, On spaces of modular forms spanned by eta-quotients, *Adv. Math.* **272** (2015), 200–224. https://arxiv.org/abs/1311.1460.

[47] D. Savitt, *An elementary proof of Newman's eta-quotient theorem*, author manuscript. https://math.jhu.edu/~savitt/papers/pdfs/newman.pdf.

[48] The Stacks Project Authors, *The Stacks Project*. https://stacks.math.columbia.edu/.

[49] T. Takagi, The cuspidal class number formula for the modular curves $X_0(M)$ with M square-free, *J. Algebra* **193** (1997), no. 1, 180–213.

[50] J. Tate and F. Oort, Group schemes of prime order, *Ann. Sci. École Norm. Sup.* (4) **3** (1970), no. 1, 1–21. https://doi.org/10.24033/asens.1186.

[51] V. Vatsal, Multiplicative subgroups of $J_0(N)$ and applications to elliptic curves, *J. Inst. Math. Jussieu* **4** (2005), no. 2, 281–316. https://doi.org/10.1017/S147474800500006X. Author version: https://personal.math.ubc.ca/~vatsal/research/optimal6.pdf.

[52] L. Wang and Y. Yang, Modular units and cuspidal divisor classes on $X_0(n^2M)$ with $n \mid 24$ and M squarefree, *J. Algebra* **562** (2020), 410–432. https://doi.org/10.1016/j.jalgebra.2020.05.041.

[53] K. Wohlfahrt, An extension of F. Klein's level concept, *Illinois J. Math.* **8** (1964), 529–535.

[54] H. Yin, A proof of the $4, 7$ cases of Sylvester's conjecture on cube sums, arXiv:2605.25917v3, 2026. https://arxiv.org/abs/2605.25917v3.

[55] H. Yoo, The rational torsion subgroup of $J_0(N)$, *Adv. Math.* **426** (2023), 109111. https://doi.org/10.1016/j.aim.2023.109111.

[56] H. Yoo, The rational cuspidal divisor class group of $X_0(N)$, *J. Number Theory* **242** (2023), 278–401. https://doi.org/10.1016/j.jnt.2022.04.009.

[57] H. Yoo and M. Yu, The rational cuspidal subgroup of $J_0(N)$, *Trans. Amer. Math. Soc.*, to appear; arXiv:2504.12564v1, 2025. https://arxiv.org/abs/2504.12564v1.

[58] D. Zywina, *On the possible images of the mod ℓ representations associated to elliptic curves over* $\mathbf{Q}$, arXiv:1508.07660, 2015. https://arxiv.org/abs/1508.07660.