In [1]:
```
import time
import sympy
import numpy as np
from math import comb
from collections import defaultdict
```

Step 1 of Edwards' Algorithm

The following computes the bound

In [2]:
```
def compute_B(r,d):
    B = 0
    if r == 3:
        B = 2*sqrt(3)*abs(d)^(1/3)
    elif r == 4:
        B = 16*sqrt(abs(d))
    elif r == 5:
        B = 3578*abs(d)
    else:
        raise Exception ("r needs to be in [3,4,5]")
    return floor(B)
```

Assume a_0 is nonzero. The following computes a_3 given a_0, a_1, a_2, and X.

In [3]:
```
def compute_a3(v_a0,v_a1,v_a2,X):
    a0, a1, a2, a3, x = var('a0, a1, a2, a3, x')
    x=X
    a0=v_a0
    a1=v_a1
    a2=v_a2
    result_a3 = sympy.solve([a0^2*a3-3*a0*a1*a2+2*a1^3==2*x, x==X, a0==v_a0,a1=
    v_a3=list(result_a3.values())[0]

    return(v_a3)
```

Assume a_0 is zero. The following computes a_3 given a_1, a_2, and X.

In [4]:
```
def compute_a3_when_a0_is_0(v_a1,v_a2):
    return 3*v_a2^2/(4*v_a1)
```

Given a_0, a_1, a_3, d, the following code computes the rest of the coefficients of the Klein form

In [5]:
```
def tetrahedral_solution_getter(result):
    v_a4 = result.get(sympy.var('a4'))
    return v_a4
```

```python
def octahedral_solution_getter(result):
    v_a4 = result[0].get(sympy.var('a4'))
    v_a5 = result[0].get(sympy.var('a5'))
    v_a6 = result[0].get(sympy.var('a6'))
    return [v_a4, v_a5, v_a6]

def tetrahedral_compute_coefficients(v_a0,v_a1,v_a2,v_a3,v_d):

    '''
    WARNING: in Edwards' PhD Thesis, the defining equations in appendix A
    has the value of d negated!!!!!
    '''

    a0, a1, a2, a3, a4,  d = var('a0, a1, a2, a3, a4, d')
    a0 = v_a0
    a1 = v_a1
    a2 = v_a2
    a3 = v_a3
    d = v_d
    eq1 = a0*a4-4*a1*a3+3*a2^2 == 0
    eq2 = a0*a2*a4+2*a1*a2*a3-a2^3-a0*a3^2-a1^2*a4 -4*d == 0

    sympy_result = sympy.solve([eq1, eq2, d == v_d, a0 == v_a0, a1 == v_a1, a2
    return tetrahedral_solution_getter(sympy_result)

def octahedral_compute_coefficients(v_a0,v_a1,v_a2,v_a3,v_d):

    '''
    WARNING: in Edwards' PhD Thesis, the defining equations in appendix A
    has the value of d negated!!!!!
    '''

    a0, a1, a2, a3, a4, a5, a6, d = var('a0, a1, a2, a3, a4, a5, a6, d')
    a0 = v_a0
    a1 = v_a1
    a2 = v_a2
    a3 = v_a3
    d = v_d

    eq1 = 0 == a0*a4 - 4*a1*a3 + 3*a2^2
    eq2 = 0 == a0*a5 - 3*a1*a4 + 2*a2*a3
    eq3 = 0 == a0*a6 - 9*a2*a4 + 8*a3^2
    eq4 = 0 == a1*a6 - 3*a2*a5 + 2*a3*a4
    eq5 = 0 == a2*a6 - 4*a3*a5 + 3*a4^2
    eq6 = 0 == -72*d + a0*a6 - 6*a1*a5 + 15*a2*a4 - 10*a3^2


    sympy_result = sympy.solve([eq1, eq2, eq3, eq4, eq5, eq6, d == v_d, a0 == v

    return octahedral_solution_getter(sympy_result)
```

The following code computes whether the coefficients satisfy the bounds

In [6]:
```python
def bound_correct(coeff, B):
    k = len(coeff) - 1
```

```
    B_squared = B * B
    for i in range(k+1):
        for j in range(k+1):
            if i+j <= k:
                if abs(coeff[i]*coeff[j]) > B_squared:
                    return False
    return True
```

In [7]:
```
def Step_1_Klein_forms(r,d):
    B = compute_B(r,d)
    resulting_forms = []

    for v_a0 in range(-B,B+1):
        for v_a1 in range(-B,B+1):
            for v_a2 in range(-B,B+1):
                Z = v_a0
                Y = v_a0*v_a2-v_a1^2
                x_abs = sqrt(-Y^3-d*(Z^r))

                # Throw out the case when X is not in Z
                if x_abs not in ZZ:
                    continue

                for X in [-x_abs,x_abs]:


                    # a0, a1 cannot be simultaneously zero
                    if v_a0 == 0 and v_a1 == 0:
                        continue

                    # compute a3. The method is different when a0 is 0 versus v
                    v_a3 = compute_a3_when_a0_is_0(v_a1,v_a2) if v_a0 == 0 else


                    #check if a3 is an integer
                    if not v_a3.is_integer:
                        continue
                    if not float(v_a3) == floor(v_a3):
                        continue

                    v_a3 = int(v_a3)
                    v_d = d

                    # Tetrahedral case
                    if r ==3:
                        v_a4 = (tetrahedral_compute_coefficients(v_a0,v_a1,v_a2
                        # DO INTEGRAL CHECK HERE. Note that integral for r=5 is
                        # In the r=5 case, we need to check that 7*v_a6 is an i

                        if not v_a4.is_integer:
                            continue

                        # Check bounds in theorem 4.3.4. of Edwards
                        coefficients = [v_a0,v_a1,v_a2,v_a3,v_a4]
                        if not bound_correct(coefficients, B):
                            continue
```

```
                        resulting_forms.append(coefficients)

                    # Octahedral case
                    elif r == 4:
                        coefficients = (octahedral_compute_coefficients(v_a0,v_

                        # DO INTEGRAL CHECK HERE. Note that integral for a5 is
                        # to check that 7*v_a6 is an integer

                        if not all(coefficient.is_integer for coefficient in co
                            continue

                        # check bounds in theorem 4.3.4
                        coefficients = [v_a0,v_a1,v_a2,v_a3]+coefficients
                        if not bound_correct(coefficients, B):
                            continue

                        resulting_forms.append(coefficients)
                    elif r == 5:
                        raise NotImplemented
                    else:
                        raise Exception ("r needs to be in [3,4,5]")

                    pass

    return resulting_forms
```

Step 2 of Edwards' Algorithm

In [ ]:

```

```

The following code multiplies each coefficient by the corresponding binomial coefficients

In [8]:

```
def return_poly_coeff_6_form(form):
    return [comb(6,i)*j for i,j in enumerate(form)]
```

In [9]:

```
def return_poly_coeff_4_form(form):
    return [comb(4,i)*j for i,j in enumerate(form)]
```

Given a 6 form, the following code check whether the the signature is $(4, 1)$

In [10]:

```
def signature_4_1(form):
    roots = np.roots(return_poly_coeff_6_form(form))
    num_real_roots = sum(np.imag(root) == 0 for root in roots)
    return (len(roots) - num_real_roots) == 2
```

Given a 6 form, the following code check whether the the signature is $(2, 2)$

In [11]:

```
def signature_2_2(form):
    roots = np.roots(return_poly_coeff_6_form(form))
    num_real_roots = sum(np.imag(root) == 0 for root in roots)
```

```
    return (len(roots) - num_real_roots) == 4
```

Given a 6 form of signature (4,1), it computes its representative point

In [12]:
```
def octahedral_4_1_rep_point(form):

    roots = np.roots(return_poly_coeff_6_form(form))

    rep_point = 0
    for candidate in roots:
        if np.imag(candidate)!= 0:
            rep_point = candidate
            break
    if np.imag(rep_point) < 0:
        rep_point = np.conj(rep_point)

    return rep_point
```

In [ ]:

In [ ]:

Given a 6 form of signature (2,2), the following computes its representative point

In [13]:
```
def compute_repn_point_4_form(roots, weights_square):
    a0_rep = sum(weights_square)
    a1_rep = sum(weights_square[i]*(roots[i]+np.conj(roots[i]))*(-1) for i in r
    a2_rep = sum(weights_square[i]*(roots[i]*np.conj(roots[i])) for i in range(

    roots = (np.roots([a0_rep,a1_rep ,a2_rep ]))

    rep_point = roots[0]
    if rep_point.imag < 0:
        rep_point = np.conj(rep_point)

    return rep_point

def octahedral_2_2_rep_point(form):
    roots = np.roots(return_poly_coeff_6_form(form))
    beta1 = 0
    beta2 = 0
    for candidate in roots:
        if np.imag(candidate)!= 0:
            beta1 = candidate
            break
    for candidate in roots:
        imag = np.imag(candidate)
        if imag!= 0 and candidate!= beta1 and imag != -np.imag(beta1):
            beta2 = candidate
            break


    beta1_bar = np.conj(beta1)
```

```
    beta2_bar = np.conj(beta2)
    t1_squared = (np.abs(beta2-beta2_bar))
    t2_squared = (np.abs(beta1-beta1_bar))

    roots_test = [beta1, beta1_bar, beta2, beta2_bar]

    weights_square_test = [t1_squared,t1_squared,t2_squared,t2_squared]
    return compute_repn_point_4_form(roots_test,weights_square_test)
```

Check whether the representative point is in the fundamental region

In [14]:
```
def in_fund_reg(rep_point):
    tol = 1e-08
    return (-1/2-tol <= np.real(rep_point) <= 1/2+tol) and np.abs(rep_point)>=1
```

Given a 4 form, the following checks whether its signature is (2,1).

In [15]:
```
def signature_2_1(form):
    roots = np.roots(return_poly_coeff_4_form(form))
    num_real_roots = sum(np.imag(root) == 0 for root in roots)
    return (len(roots) - num_real_roots) == 2
```

Given a 4 form of signature (2,1), the following computes its representative point

In [16]:
```
def tetrahedral_2_1_rep_point(form):
    roots = np.roots(return_poly_coeff_4_form(form))

    num_real_roots = sum(np.imag(root) == 0 for root in roots)

    assert  num_real_roots == 2

    real_roots = [root for root in roots if np.imag(root) == 0 ]
    complex_roots = [root for root in roots if np.imag(root) != 0 ]

    assert(len(real_roots) == 2)

    alpha1, alpha2 = real_roots
    beta,beta_bar = complex_roots

    t1_squared = np.abs((beta-beta_bar)*(alpha2-beta)^2)
    t2_squared = np.abs((beta-beta_bar)*(alpha1-beta)^2)
    u_squared = np.abs((alpha1-alpha2)*(alpha1-beta)*(alpha2-beta))


    roots_test = [alpha1, alpha2, beta, beta_bar]

    weights_square_test = [t1_squared,t2_squared,u_squared,u_squared]

    return compute_repn_point_4_form(roots_test,weights_square_test)
```

In [ ]:
```
```

The following computes transformation of a form by T, S, U of GL2(Z)

```
In [17]: def T_transformation(original_form):
    new_form = [0] * len(original_form)
    for index in range(len(original_form)):
        new_form[index] = original_form[index]
        new_form[index] += sum(comb(index, i)*original_form[i] for i in range(:
    return new_form
```

```
In [18]: def U_transformation(original_form):
    return [coeff*(-1)**index for index,coeff in enumerate(original_form)]
```

```
In [19]: def S_transformation(original_form):
    return [coeff*(-1)**index for index,coeff in enumerate(original_form[::-1])
```

```
In [20]: def US_transformation(original_form):
    return original_form[::-1]
```

```
In [21]: def ST_transformation(original_form):
    return S_transformation(T_transformation(original_form))
```

```
In [ ]:
```

Given a list of (form, representative_point) tuple, the following rounds the representative point to the nearest 8 decimap places. i.e. 0.4999999999999999 becomes 0.5

```
In [22]: def clean_up(results):

    deduped = list(set((tuple(form), np.around(point, decimals=8, out=None)) fo
    deduped.sort()
    return deduped

```

The following code returns a tuple. The first tuple is whether the representative point is in fundamental region. The second is the representative point.

```
In [23]: def repn_point_in_fund_region(r, coefficients):
    if r == 3:
        rep_point = 0
        # determine the representative point when a0=0 (Tetrahedral case)
        if coefficients[0] == 0:
            modified_coefficients = coefficients[::1]
            # transform by T until one of a0 and a5 is nonzero.

            while modified_coefficients[-1] == 0:
                modified_coefficients = T_transformation(modified_coefficients)
            modified_coefficients = modified_coefficients[::-1]

            # then reverse the coefficients and take reciprocal of the point
            rep_point = 1/(tetrahedral_2_1_rep_point(modified_coefficients))
            if np.imag(rep_point)<0:
                rep_point = np.conj(rep_point)
        else:
            # check signature and compute repn points
            roots = np.roots(return_poly_coeff_4_form(coefficients))
```

```
            if signature_2_1(coefficients):
                rep_point = tetrahedral_2_1_rep_point(coefficients)

            else: # Note that (4,0) and (0,2) have not been encountered. (0,2)
                raise NotImplemented

        # return false when the representative point is not in fundamental regi
        return (in_fund_reg(rep_point), rep_point)


    elif r == 4:
        # computing the rep point depending on the signature
        rep_point = 0
        if signature_4_1(coefficients):
            rep_point = octahedral_4_1_rep_point(coefficients)


        elif signature_2_2(coefficients):
            rep_point = octahedral_2_2_rep_point(coefficients)

        else: # signature must be of 4,1 or 2,2.
              #Signature of 3,0 cannot be a form in C(r,d)
            raise NotImplemented
        # return false when the representative point is not in fundamental regi
        return (in_fund_reg(rep_point), rep_point)

    elif r == 5: #r == 5
        raise NotImplemented
    else:
        raise Exception ("r needs to be in [3,4,5]")
```

In [ ]:

In [ ]:

In [ ]:

The following code implements step 1 of Edwards' algorithm. It is the most computationally heavy step

In [24]:
```
def Step_2_Hermite_reduced_forms(r, forms):

    resulting_forms = []
    for coefficients in forms:
        in_region, rep_point = repn_point_in_fund_region(r, coefficients)

        if in_region:
            resulting_forms.append((coefficients,rep_point))

    return clean_up(resulting_forms)
```

Step 3 of Edwards' Algorithm

Following throws away forms whose representative point is not GL2(Z) reduced

In [25]:
```
def keep_gl2z_reduced_forms(results):
    return [(form, point) for form, point in results if np.real(point)<=0]
```

The given a list of forms, generates a list of forms that contains each form and the negatives of each form

In [26]:
```
def generate_minus_I(forms):

    return forms + [[item*-1 for item in some_list] for some_list in forms]
```

The following code implements lemma 5.2.1. That is, given a form, this generates a list of all forms that is in its GL2Z orbit

In [27]:
```
def gl2_equiv_forms(form, rep_point):

    omega = -0.5+0.8660254j

    if np.isclose(omega, rep_point):
        ST_orbit = [form, ST_transformation(form), ST_transformation(ST_transfo
        ST_T_orbit = [US_transformation(orb) for orb in ST_orbit]
        return generate_minus_I(ST_orbit+ST_T_orbit)


    else:
        if np.real(rep_point)== 0 :
            return generate_minus_I([form, U_transformation(form)])

        elif np.isclose(np.abs(rep_point) ,1):
            return generate_minus_I([form,  US_transformation(form)])

        elif np.real(rep_point) == -1/2:
            return generate_minus_I([form, U_transformation(form)])

        else:
            return generate_minus_I([form])
```

Given list of forms, this code groups the forms together if they have the same GL2(Z) orbit

In [28]:
```
def put_into_gl2z_equivalence(results):

    # The keys are forms, representing a set of gl_2(Z) equivalent forms
    # the values are a list of form, rep_point pairs

    lookup = defaultdict(list)

    print("here are the GL2Z equivalent classes")

    for form, rep_point in results:
        gl2_equivalent = gl2_equiv_forms(list(form), rep_point)
        key = min(gl2_equivalent)

        lookup[tuple(key)].append((form, rep_point))
```

```
    class_representatives = []
    for key, value in lookup.items():
        class_representatives.append(value[0])
        print('Following forms are in same orbit')
        for form, rep_point in value:
            print('----', form, rep_point)

    return class_representatives
```

In [29]:
```
def Algorithm_step_3(results):
    resulting_forms_and_points = []

    return put_into_gl2z_equivalence(keep_gl2z_reduced_forms(results))
```

Step 4 of Edwards' Algorithm

The following computes the Hessian of a form f, which is a covariant

In [30]:
```
def H(f,k):
    x,y = sympy.symbols('x,y')
    fx = sympy.diff(f, x)
    fy = sympy.diff(f, y)
    fxx = sympy.diff(fx,x)
    fyy = sympy.diff(fy,y)
    fxy = sympy.diff(fx,y)
    fyx = sympy.diff(fy,x)

    return (fxx*fyy-fxy*fyx)*1/((k*(k-1))^2)
```

The following computes the functional determinant of a form f, which is also a covariant

In [31]:
```
def t(f,H,k):
    x,y = sympy.symbols('x,y')
    fx = sympy.diff(f, x)
    fy = sympy.diff(f, y)
    Hx = sympy.diff(H,x)
    Hy = sympy.diff(H,y)
    return (fx*Hy-fy*Hx)/(k*(k-2))
```

The following code produces (x, y, z) as an integer specialization of $\mathscr{C}(r, d)$ by s_1, s_2

In [32]:
```
def get_xyz_from_s1s2(t,H,f,s1,s2):
    x,y = sympy.symbols('x,y')
    t_number = t.subs({x:s1, y: s2})
    H_number = H.subs({x:s1, y: s2})
    f_number = f.subs({x:s1, y: s2})


    return (t_number/2,H_number, f_number)
```

Determines N_0 from N and d

In [33]:
```
def determine_N0(N,d):
    Nd = N*d
    N0 = 1
    for p in primes(Nd):
        if p!= 2 and p.divides(Nd):
            N0 *= p

    return N0
```

The following checks whether a form generates coprime solutions

In [34]:
```
def step_4_check_coprime(form, d):
    x,y = sympy.symbols('x,y')
    k = len(form)-1
    f_v = sum(comb(k,i)*form[i]*x**i*y**(k-i) for i in range(k+1))
    H_v = H(f_v,k)
    t_v = t(f_v,H_v,k)
    N = 0
    if k == 4:
        N = 12
    elif k == 6:
        N = 24
    else:
        raise NotImplemented

    N0 = determine_N0(N,d)
    for s1 in range(-N0, N0+1):
        for s2 in range(-N0, N0+1):
            x_v,y_v,z_v = get_xyz_from_s1s2(t_v,H_v,f_v,s1,s2)
            if gcd(gcd(x_v,y_v),z_v) == 1:
                return True
    return False
```

In [35]:
```
def step_4(forms, d):
    results = []
    for form, rep_point in forms:
        if step_4_check_coprime(form,d):
            results.append((form, rep_point))
    return results
```

The following algorithm puts step 1, step 2, step 3 together

In [36]:
```
def whole_algo(r,d):
    print('-------------------------------------------')
    print('r,d is ', r,d)
    step_1_results = Step_1_Klein_forms(r,d)

    step_2_results = Step_2_Hermite_reduced_forms(r, step_1_results)
    print('step_1_2_combined_results:')
    for tup in step_2_results:
```

```
        print(tup)

    print('step_3_results:')
    step_3_results = Algorithm_step_3(step_2_results)

    print('step_4_results:')
    step_4_results = step_4(step_3_results,r)
    for result in step_4_results:
        print(result)

    return  step_4_results
```

Below code runs the whole algorithm on several different parameters

In [40]:
```
parameters_to_run = [[3,1],[3,-1],[3,2],[3,-2],[3,3],[3,-3],[3,4],[3,-4],[4,1],
```

In [38]:
```
results = []
for r,d in parameters_to_run:
    result = whole_algo(r,d)
    results.append(result)
```

```
------------------------------------------
r,d is  3 1
step_1_2_combined_results:
((-2, -1, 0, -1, -2), (-0.26794919+0.96343304j))
((-2, 1, 0, 1, -2), (0.26794919+0.96343304j))
((-1, -1, 1, -1, -1), (0.26794919+0.96343304j))
((-1, 0, -1, 0, 3), (-0+1.31607401j))
((-1, 0, 0, -2, 0), 1.41421356j)
((-1, 0, 0, 2, 0), 1.41421356j)
((-1, 1, 1, 1, -1), (-0.26794919+0.96343304j))
((0, -1, 0, 0, -4), 1.41421356j)
((0, -1, 2, -3, 0), 1.41421356j)
((0, 1, 0, 0, -4), 1.41421356j)
((1, 0, -1, 0, -3), (-0+1.31607401j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-2, -1, 0, -1, -2) (-0.26794919+0.96343304j)
Following forms are in same orbit
---- (-1, 0, -1, 0, 3) (-0+1.31607401j)
Following forms are in same orbit
---- (-1, 0, 0, -2, 0) 1.41421356j
---- (-1, 0, 0, 2, 0) 1.41421356j
Following forms are in same orbit
---- (-1, 1, 1, 1, -1) (-0.26794919+0.96343304j)
Following forms are in same orbit
---- (0, -1, 0, 0, -4) 1.41421356j
---- (0, 1, 0, 0, -4) 1.41421356j
Following forms are in same orbit
---- (0, -1, 2, -3, 0) 1.41421356j
Following forms are in same orbit
---- (1, 0, -1, 0, -3) (-0+1.31607401j)
step_4_results:
((-2, -1, 0, -1, -2), (-0.26794919+0.96343304j))
((-1, 0, -1, 0, 3), (-0+1.31607401j))
((-1, 0, 0, -2, 0), 1.41421356j)
((-1, 1, 1, 1, -1), (-0.26794919+0.96343304j))
((0, -1, 0, 0, -4), 1.41421356j)
((0, -1, 2, -3, 0), 1.41421356j)
((1, 0, -1, 0, -3), (-0+1.31607401j))
------------------------------------------
r,d is  3 -1
step_1_2_combined_results:
((-1, 0, 1, 0, 3), (-0+1.31607401j))
((0, -1, 0, 0, 4), 1.41421356j)
((0, 1, -2, 3, 0), 1.41421356j)
((0, 1, 0, 0, 4), 1.41421356j)
((1, -1, -1, -1, 1), (-0.26794919+0.96343304j))
((1, 0, 0, -2, 0), 1.41421356j)
((1, 0, 0, 2, 0), 1.41421356j)
((1, 0, 1, 0, -3), (-0+1.31607401j))
((1, 1, -1, 1, 1), (0.26794919+0.96343304j))
((2, -1, 0, -1, 2), (0.26794919+0.96343304j))
((2, 1, 0, 1, 2), (-0.26794919+0.96343304j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
```

```
---- (-1, 0, 1, 0, 3) (-0+1.31607401j)
Following forms are in same orbit
---- (0, -1, 0, 0, 4) 1.41421356j
---- (0, 1, 0, 0, 4) 1.41421356j
Following forms are in same orbit
---- (0, 1, -2, 3, 0) 1.41421356j
Following forms are in same orbit
---- (1, -1, -1, -1, 1) (-0.26794919+0.96343304j)
Following forms are in same orbit
---- (1, 0, 0, -2, 0) 1.41421356j
---- (1, 0, 0, 2, 0) 1.41421356j
Following forms are in same orbit
---- (1, 0, 1, 0, -3) (-0+1.31607401j)
Following forms are in same orbit
---- (2, 1, 0, 1, 2) (-0.26794919+0.96343304j)
step_4_results:
((-1, 0, 1, 0, 3), (-0+1.31607401j))
((0, -1, 0, 0, 4), 1.41421356j)
((0, 1, -2, 3, 0), 1.41421356j)
((1, -1, -1, -1, 1), (-0.26794919+0.96343304j))
((1, 0, 0, -2, 0), 1.41421356j)
((1, 0, 1, 0, -3), (-0+1.31607401j))
((2, 1, 0, 1, 2), (-0.26794919+0.96343304j))
------------------------------------------
r,d is  3 2
step_1_2_combined_results:
((-2, 0, 0, -2, 0), 1.12246205j)
((-2, 0, 0, 2, 0), 1.12246205j)
((-1, 0, -1, -2, 3), (-0.15826276+1.50897958j))
((-1, 0, -1, 2, 3), (0.15826276+1.50897958j))
((0, -1, 0, 0, -8), 1.78179744j)
((0, 1, 0, 0, -8), 1.78179744j)
((1, -2, 1, 0, -3), (0.43612414+1.0108584j))
((1, 2, 1, 0, -3), (-0.43612414+1.0108584j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-2, 0, 0, -2, 0) 1.12246205j
---- (-2, 0, 0, 2, 0) 1.12246205j
Following forms are in same orbit
---- (-1, 0, -1, -2, 3) (-0.15826276+1.50897958j)
Following forms are in same orbit
---- (0, -1, 0, 0, -8) 1.78179744j
---- (0, 1, 0, 0, -8) 1.78179744j
Following forms are in same orbit
---- (1, 2, 1, 0, -3) (-0.43612414+1.0108584j)
step_4_results:
((-1, 0, -1, -2, 3), (-0.15826276+1.50897958j))
((0, -1, 0, 0, -8), 1.78179744j)
((1, 2, 1, 0, -3), (-0.43612414+1.0108584j))
------------------------------------------
r,d is  3 -2
step_1_2_combined_results:
((-1, -2, -1, 0, 3), (-0.43612414+1.0108584j))
((-1, 2, -1, 0, 3), (0.43612414+1.0108584j))
((0, -1, 0, 0, 8), 1.78179744j)
((0, 1, 0, 0, 8), 1.78179744j)
```

```
((1, 0, 1, -2, -3), (0.15826276+1.50897958j))
((1, 0, 1, 2, -3), (-0.15826276+1.50897958j))
((2, 0, 0, -2, 0), 1.12246205j)
((2, 0, 0, 2, 0), 1.12246205j)
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-1, -2, -1, 0, 3) (-0.43612414+1.0108584j)
Following forms are in same orbit
---- (0, -1, 0, 0, 8) 1.78179744j
---- (0, 1, 0, 0, 8) 1.78179744j
Following forms are in same orbit
---- (1, 0, 1, 2, -3) (-0.15826276+1.50897958j)
Following forms are in same orbit
---- (2, 0, 0, -2, 0) 1.12246205j
---- (2, 0, 0, 2, 0) 1.12246205j
step_4_results:
((-1, -2, -1, 0, 3), (-0.43612414+1.0108584j))
((0, -1, 0, 0, 8), 1.78179744j)
((1, 0, 1, 2, -3), (-0.15826276+1.50897958j))
------------------------------------------
r,d is  3 3
step_1_2_combined_results:
((-2, -1, 0, -2, -4), (-0.31822819+1.18033017j))
((-2, 1, 0, 2, -4), (0.31822819+1.18033017j))
((-1, -1, 0, -2, -8), (-0.42588044+1.57961975j))
((-1, 1, 0, 2, -8), (0.42588044+1.57961975j))
((0, -2, 0, 0, -3), 1.01982445j)
((0, -1, 0, 0, -12), 2.0396489j)
((0, 1, 0, 0, -12), 2.0396489j)
((0, 2, 0, 0, -3), 1.01982445j)
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-2, -1, 0, -2, -4) (-0.31822819+1.18033017j)
Following forms are in same orbit
---- (-1, -1, 0, -2, -8) (-0.42588044+1.57961975j)
Following forms are in same orbit
---- (0, -2, 0, 0, -3) 1.01982445j
---- (0, 2, 0, 0, -3) 1.01982445j
Following forms are in same orbit
---- (0, -1, 0, 0, -12) 2.0396489j
---- (0, 1, 0, 0, -12) 2.0396489j
step_4_results:
((-2, -1, 0, -2, -4), (-0.31822819+1.18033017j))
((-1, -1, 0, -2, -8), (-0.42588044+1.57961975j))
((0, -2, 0, 0, -3), 1.01982445j)
((0, -1, 0, 0, -12), 2.0396489j)
------------------------------------------
r,d is  3 -3
step_1_2_combined_results:
((0, -2, 0, 0, 3), 1.01982445j)
((0, -1, 0, 0, 12), 2.0396489j)
((0, 1, 0, 0, 12), 2.0396489j)
((0, 2, 0, 0, 3), 1.01982445j)
((1, -1, 0, -2, 8), (0.42588044+1.57961975j))
((1, 1, 0, 2, 8), (-0.42588044+1.57961975j))
```

```
((2, -1, 0, -2, 4), (0.31822819+1.18033017j))
((2, 1, 0, 2, 4), (-0.31822819+1.18033017j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (0, -2, 0, 0, 3) 1.01982445j
---- (0, 2, 0, 0, 3) 1.01982445j
Following forms are in same orbit
---- (0, -1, 0, 0, 12) 2.0396489j
---- (0, 1, 0, 0, 12) 2.0396489j
Following forms are in same orbit
---- (1, 1, 0, 2, 8) (-0.42588044+1.57961975j)
Following forms are in same orbit
---- (2, 1, 0, 2, 4) (-0.31822819+1.18033017j)
step_4_results:
((0, -2, 0, 0, 3), 1.01982445j)
((0, -1, 0, 0, 12), 2.0396489j)
((1, 1, 0, 2, 8), (-0.42588044+1.57961975j))
((2, 1, 0, 2, 4), (-0.31822819+1.18033017j))
------------------------------------------
r,d is  3 4
step_1_2_combined_results:
((-3, -1, -1, -3, -3), (-0.43612414+1.0108584j))
((-3, 0, -1, -2, 1), (-0.11501333+0.99336395j))
((-3, 0, -1, 2, 1), (0.11501333+0.99336395j))
((-3, 1, -1, 3, -3), (0.43612414+1.0108584j))
((-1, 0, 0, -4, 0), 1.78179744j)
((-1, 0, 0, 4, 0), 1.78179744j)
((0, -2, 0, 0, -4), 1.12246205j)
((0, -1, 0, 0, -16), 2.2449241j)
((0, 1, 0, 0, -16), 2.2449241j)
((0, 2, 0, 0, -4), 1.12246205j)
((1, -2, -1, 0, -3), (-0.11501333+0.99336395j))
((1, -1, -1, 1, -7), (0.15826276+1.50897958j))
((1, 1, -1, -1, -7), (-0.15826276+1.50897958j))
((1, 2, -1, 0, -3), (0.11501333+0.99336395j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-3, -1, -1, -3, -3) (-0.43612414+1.0108584j)
Following forms are in same orbit
---- (-3, 0, -1, -2, 1) (-0.11501333+0.99336395j)
---- (1, -2, -1, 0, -3) (-0.11501333+0.99336395j)
Following forms are in same orbit
---- (-1, 0, 0, -4, 0) 1.78179744j
---- (-1, 0, 0, 4, 0) 1.78179744j
Following forms are in same orbit
---- (0, -2, 0, 0, -4) 1.12246205j
---- (0, 2, 0, 0, -4) 1.12246205j
Following forms are in same orbit
---- (0, -1, 0, 0, -16) 2.2449241j
---- (0, 1, 0, 0, -16) 2.2449241j
Following forms are in same orbit
---- (1, 1, -1, -1, -7) (-0.15826276+1.50897958j)
step_4_results:
((-3, -1, -1, -3, -3), (-0.43612414+1.0108584j))
((-3, 0, -1, -2, 1), (-0.11501333+0.99336395j))
```

```
((-1, 0, 0, -4, 0), 1.78179744j)
((0, -1, 0, 0, -16), 2.2449241j)
((1, 1, -1, -1, -7), (-0.15826276+1.50897958j))
------------------------------------------
r,d is  3 -4
step_1_2_combined_results:
((-1, -2, 1, 0, 3), (0.11501333+0.99336395j))
((-1, -1, 1, 1, 7), (-0.15826276+1.50897958j))
((-1, 1, 1, -1, 7), (0.15826276+1.50897958j))
((-1, 2, 1, 0, 3), (-0.11501333+0.99336395j))
((0, -2, 0, 0, 4), 1.12246205j)
((0, -1, 0, 0, 16), 2.2449241j)
((0, 1, 0, 0, 16), 2.2449241j)
((0, 2, 0, 0, 4), 1.12246205j)
((1, 0, 0, -4, 0), 1.78179744j)
((1, 0, 0, 4, 0), 1.78179744j)
((3, -1, 1, -3, 3), (0.43612414+1.0108584j))
((3, 0, 1, -2, -1), (0.11501333+0.99336395j))
((3, 0, 1, 2, -1), (-0.11501333+0.99336395j))
((3, 1, 1, 3, 3), (-0.43612414+1.0108584j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-1, -1, 1, 1, 7) (-0.15826276+1.50897958j)
Following forms are in same orbit
---- (-1, 2, 1, 0, 3) (-0.11501333+0.99336395j)
---- (3, 0, 1, 2, -1) (-0.11501333+0.99336395j)
Following forms are in same orbit
---- (0, -2, 0, 0, 4) 1.12246205j
---- (0, 2, 0, 0, 4) 1.12246205j
Following forms are in same orbit
---- (0, -1, 0, 0, 16) 2.2449241j
---- (0, 1, 0, 0, 16) 2.2449241j
Following forms are in same orbit
---- (1, 0, 0, -4, 0) 1.78179744j
---- (1, 0, 0, 4, 0) 1.78179744j
Following forms are in same orbit
---- (3, 1, 1, 3, 3) (-0.43612414+1.0108584j)
step_4_results:
((-1, -1, 1, 1, 7), (-0.15826276+1.50897958j))
((-1, 2, 1, 0, 3), (-0.11501333+0.99336395j))
((0, -1, 0, 0, 16), 2.2449241j)
((1, 0, 0, -4, 0), 1.78179744j)
((3, 1, 1, 3, 3), (-0.43612414+1.0108584j))
------------------------------------------
r,d is  4 1
step_1_2_combined_results:
((-8, -4, 0, 2, 4, 6, 4), (-0.5+0.8660254j))
((-8, 4, 0, -2, 4, -6, 4), (0.5+0.8660254j))
((-4, -6, -4, -2, 0, 4, 8), (-0.5+0.8660254j))
((-4, -2, 4, -4, 4, -2, -4), (0.5+0.8660254j))
((-4, 2, 4, 4, 4, 2, -4), (-0.5+0.8660254j))
((-4, 6, -4, 2, 0, -4, 8), (0.5+0.8660254j))
((-3, -4, -1, 0, 1, 4, 3), (-0.26794919+0.96343304j))
((-3, 4, -1, 0, 1, -4, 3), (0.26794919+0.96343304j))
((-1, 0, 1, 0, 3, 0, -27), (-0+1.73205081j))
((0, -3, 0, 0, 0, 4, 0), 1.07456993j)
```

```
((0, -2, 0, 0, 0, 6, 0), (-0+1.31607401j))
((0, -1, 0, 0, 0, 12, 0), 1.86120972j)
((0, 1, 0, 0, 0, -12, 0), 1.86120972j)
((0, 2, 0, 0, 0, -6, 0), (-0+1.31607401j))
((0, 3, 0, 0, 0, -4, 0), 1.07456993j)
((1, 0, -1, 0, -3, 0, 27), (-0+1.73205081j))
((3, -4, 1, 0, -1, 4, -3), (0.26794919+0.96343304j))
((3, 4, 1, 0, -1, -4, -3), (-0.26794919+0.96343304j))
((4, -6, 4, -2, 0, 4, -8), (0.5+0.8660254j))
((4, -2, -4, -4, -4, -2, 4), (-0.5+0.8660254j))
((4, 2, -4, 4, -4, 2, 4), (0.5+0.8660254j))
((4, 6, 4, 2, 0, -4, -8), (-0.5+0.8660254j))
((8, -4, 0, 2, -4, 6, -4), (0.5+0.8660254j))
((8, 4, 0, -2, -4, -6, -4), (-0.5+0.8660254j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-8, -4, 0, 2, 4, 6, 4) (-0.5+0.8660254j)
---- (-4, -6, -4, -2, 0, 4, 8) (-0.5+0.8660254j)
---- (4, 6, 4, 2, 0, -4, -8) (-0.5+0.8660254j)
---- (8, 4, 0, -2, -4, -6, -4) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-4, 2, 4, 4, 4, 2, -4) (-0.5+0.8660254j)
---- (4, -2, -4, -4, -4, -2, 4) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-3, -4, -1, 0, 1, 4, 3) (-0.26794919+0.96343304j)
---- (3, 4, 1, 0, -1, -4, -3) (-0.26794919+0.96343304j)
Following forms are in same orbit
---- (-1, 0, 1, 0, 3, 0, -27) (-0+1.73205081j)
---- (1, 0, -1, 0, -3, 0, 27) (-0+1.73205081j)
Following forms are in same orbit
---- (0, -3, 0, 0, 0, 4, 0) 1.07456993j
---- (0, 3, 0, 0, 0, -4, 0) 1.07456993j
Following forms are in same orbit
---- (0, -2, 0, 0, 0, 6, 0) (-0+1.31607401j)
---- (0, 2, 0, 0, 0, -6, 0) (-0+1.31607401j)
Following forms are in same orbit
---- (0, -1, 0, 0, 0, 12, 0) 1.86120972j
---- (0, 1, 0, 0, 0, -12, 0) 1.86120972j
step_4_results:
((-3, -4, -1, 0, 1, 4, 3), (-0.26794919+0.96343304j))
((-1, 0, 1, 0, 3, 0, -27), (-0+1.73205081j))
((0, -3, 0, 0, 0, 4, 0), 1.07456993j)
((0, -1, 0, 0, 0, 12, 0), 1.86120972j)
------------------------------------------
r,d is  4 -1
step_1_2_combined_results:
((-8, -4, -4, -4, -2, 1, 7), (-0.5+0.8660254j))
((-8, 4, -4, 4, -2, -1, 7), (0.5+0.8660254j))
((-7, -6, -3, -2, -3, -6, -7), (-0.5+0.8660254j))
((-7, -1, 2, 4, 4, 4, 8), (-0.5+0.8660254j))
((-7, 1, 2, -4, 4, -4, 8), (0.5+0.8660254j))
((-7, 6, -3, 2, -3, 6, -7), (0.5+0.8660254j))
((-6, -2, -2, 0, 2, 2, 6), (-0.26794919+0.96343304j))
((-6, 2, -2, 0, 2, -2, 6), (0.26794919+0.96343304j))
((-5, -1, 1, 3, 3, 3, 9), (-0.43612414+1.0108584j))
((-5, 1, 1, -3, 3, -3, 9), (0.43612414+1.0108584j))
```

```
((-4, 0, 0, -2, 0, 0, 8), 1.12246205j)
((-4, 0, 0, 2, 0, 0, 8), (-0+1.12246205j))
((-2, 0, 0, -2, 0, 0, 16), (-0+1.41421356j))
((-2, 0, 0, 2, 0, 0, 16), 1.41421356j)
((-1, -1, 1, -1, -1, -5, 17), (0.15826276+1.50897958j))
((-1, 0, -1, 0, 3, 0, 27), 1.73205081j)
((-1, 0, 0, -2, 0, 0, 32), (-0+1.78179744j))
((-1, 0, 0, 2, 0, 0, 32), 1.78179744j)
((-1, 1, 1, 1, -1, 5, 17), (-0.15826276+1.50897958j))
((0, -3, 0, 0, 0, -4, 0), 1.07456993j)
((0, -2, 0, 0, 0, -6, 0), 1.31607401j)
((0, -1, 0, 0, 0, -12, 0), (-0+1.86120972j))
((0, 1, 0, 0, 0, 12, 0), (-0+1.86120972j))
((0, 2, 0, 0, 0, 6, 0), 1.31607401j)
((0, 3, 0, 0, 0, 4, 0), 1.07456993j)
((1, -1, -1, -1, 1, -5, -17), (-0.15826276+1.50897958j))
((1, 0, 0, -2, 0, 0, -32), 1.78179744j)
((1, 0, 0, 2, 0, 0, -32), (-0+1.78179744j))
((1, 0, 1, 0, -3, 0, -27), 1.73205081j)
((1, 1, -1, 1, 1, 5, -17), (0.15826276+1.50897958j))
((2, 0, 0, -2, 0, 0, -16), 1.41421356j)
((2, 0, 0, 2, 0, 0, -16), (-0+1.41421356j))
((4, 0, 0, -2, 0, 0, -8), (-0+1.12246205j))
((4, 0, 0, 2, 0, 0, -8), 1.12246205j)
((5, -1, -1, 3, -3, 3, -9), (0.43612414+1.0108584j))
((5, 1, -1, -3, -3, -3, -9), (-0.43612414+1.0108584j))
((6, -2, 2, 0, -2, 2, -6), (0.26794919+0.96343304j))
((6, 2, 2, 0, -2, -2, -6), (-0.26794919+0.96343304j))
((7, -6, 3, -2, 3, -6, 7), (0.5+0.8660254j))
((7, -1, -2, 4, -4, 4, -8), (0.5+0.8660254j))
((7, 1, -2, -4, -4, -4, -8), (-0.5+0.8660254j))
((7, 6, 3, 2, 3, 6, 7), (-0.5+0.8660254j))
((8, -4, 4, -4, 2, 1, -7), (0.5+0.8660254j))
((8, 4, 4, 4, 2, -1, -7), (-0.5+0.8660254j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-8, -4, -4, -4, -2, 1, 7) (-0.5+0.8660254j)
---- (-7, -1, 2, 4, 4, 4, 8) (-0.5+0.8660254j)
---- (7, 1, -2, -4, -4, -4, -8) (-0.5+0.8660254j)
---- (8, 4, 4, 4, 2, -1, -7) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-7, -6, -3, -2, -3, -6, -7) (-0.5+0.8660254j)
---- (7, 6, 3, 2, 3, 6, 7) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-6, -2, -2, 0, 2, 2, 6) (-0.26794919+0.96343304j)
---- (6, 2, 2, 0, -2, -2, -6) (-0.26794919+0.96343304j)
Following forms are in same orbit
---- (-5, -1, 1, 3, 3, 3, 9) (-0.43612414+1.0108584j)
---- (5, 1, -1, -3, -3, -3, -9) (-0.43612414+1.0108584j)
Following forms are in same orbit
---- (-4, 0, 0, -2, 0, 0, 8) 1.12246205j
---- (-4, 0, 0, 2, 0, 0, 8) (-0+1.12246205j)
---- (4, 0, 0, -2, 0, 0, -8) (-0+1.12246205j)
---- (4, 0, 0, 2, 0, 0, -8) 1.12246205j
Following forms are in same orbit
---- (-2, 0, 0, -2, 0, 0, 16) (-0+1.41421356j)
```

```
---- (-2, 0, 0, 2, 0, 0, 16) 1.41421356j
---- (2, 0, 0, -2, 0, 0, -16) 1.41421356j
---- (2, 0, 0, 2, 0, 0, -16) (-0+1.41421356j)
Following forms are in same orbit
---- (-1, 0, -1, 0, 3, 0, 27) 1.73205081j
---- (1, 0, 1, 0, -3, 0, -27) 1.73205081j
Following forms are in same orbit
---- (-1, 0, 0, -2, 0, 0, 32) (-0+1.78179744j)
---- (-1, 0, 0, 2, 0, 0, 32) 1.78179744j
---- (1, 0, 0, -2, 0, 0, -32) 1.78179744j
---- (1, 0, 0, 2, 0, 0, -32) (-0+1.78179744j)
Following forms are in same orbit
---- (-1, 1, 1, 1, -1, 5, 17) (-0.15826276+1.50897958j)
---- (1, -1, -1, -1, 1, -5, -17) (-0.15826276+1.50897958j)
Following forms are in same orbit
---- (0, -3, 0, 0, 0, -4, 0) 1.07456993j
---- (0, 3, 0, 0, 0, 4, 0) 1.07456993j
Following forms are in same orbit
---- (0, -2, 0, 0, 0, -6, 0) 1.31607401j
---- (0, 2, 0, 0, 0, 6, 0) 1.31607401j
Following forms are in same orbit
---- (0, -1, 0, 0, 0, -12, 0) (-0+1.86120972j)
---- (0, 1, 0, 0, 0, 12, 0) (-0+1.86120972j)
step_4_results:
((-8, -4, -4, -4, -2, 1, 7), (-0.5+0.8660254j))
((-7, -6, -3, -2, -3, -6, -7), (-0.5+0.8660254j))
((-5, -1, 1, 3, 3, 3, 9), (-0.43612414+1.0108584j))
((-1, 0, -1, 0, 3, 0, 27), 1.73205081j)
((-1, 0, 0, -2, 0, 0, 32), (-0+1.78179744j))
((-1, 1, 1, 1, -1, 5, 17), (-0.15826276+1.50897958j))
((0, -3, 0, 0, 0, -4, 0), 1.07456993j)
((0, -1, 0, 0, 0, -12, 0), (-0+1.86120972j))
------------------------------------------
r,d is  4 2
step_1_2_combined_results:
((-5, -1, 2, 2, 4, 4, -8), (-0.31010205+1.13297512j))
((-5, 1, 2, -2, 4, -4, -8), (0.31010205+1.13297512j))
((-1, -1, 2, -2, 4, 4, -40), (0.44948974+1.64223581j))
((-1, 1, 2, 2, 4, -4, -40), (-0.44948974+1.64223581j))
((0, -4, 0, 0, 0, 6, 0), 1.10668192j)
((0, -3, 0, 0, 0, 8, 0), (-0+1.27788621j))
((0, -2, 0, 0, 0, 12, 0), 1.56508458j)
((0, -1, 0, 0, 0, 24, 0), 2.21336384j)
((0, 1, 0, 0, 0, -24, 0), 2.21336384j)
((0, 2, 0, 0, 0, -12, 0), 1.56508458j)
((0, 3, 0, 0, 0, -8, 0), (-0+1.27788621j))
((0, 4, 0, 0, 0, -6, 0), 1.10668192j)
((1, -1, -2, -2, -4, 4, 40), (-0.44948974+1.64223581j))
((1, 1, -2, 2, -4, -4, 40), (0.44948974+1.64223581j))
((5, -1, -2, 2, -4, 4, 8), (0.31010205+1.13297512j))
((5, 1, -2, -2, -4, -4, 8), (-0.31010205+1.13297512j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-5, -1, 2, 2, 4, 4, -8) (-0.31010205+1.13297512j)
---- (5, 1, -2, -2, -4, -4, 8) (-0.31010205+1.13297512j)
Following forms are in same orbit
```

```
---- (-1, 1, 2, 2, 4, -4, -40) (-0.44948974+1.64223581j)
---- (1, -1, -2, -2, -4, 4, 40) (-0.44948974+1.64223581j)
Following forms are in same orbit
---- (0, -4, 0, 0, 0, 6, 0) 1.10668192j
---- (0, 4, 0, 0, 0, -6, 0) 1.10668192j
Following forms are in same orbit
---- (0, -3, 0, 0, 0, 8, 0) (-0+1.27788621j)
---- (0, 3, 0, 0, 0, -8, 0) (-0+1.27788621j)
Following forms are in same orbit
---- (0, -2, 0, 0, 0, 12, 0) 1.56508458j
---- (0, 2, 0, 0, 0, -12, 0) 1.56508458j
Following forms are in same orbit
---- (0, -1, 0, 0, 0, 24, 0) 2.21336384j
---- (0, 1, 0, 0, 0, -24, 0) 2.21336384j
step_4_results:
((-5, -1, 2, 2, 4, 4, -8), (-0.31010205+1.13297512j))
((-1, 1, 2, 2, 4, -4, -40), (-0.44948974+1.64223581j))
((0, -3, 0, 0, 0, 8, 0), (-0+1.27788621j))
((0, -1, 0, 0, 0, 24, 0), 2.21336384j)
------------------------------------------
r,d is  4 -2
step_1_2_combined_results:
((-8, -4, -4, -2, 2, 5, 13), (-0.43771272+0.98025439j))
((-8, 4, -4, 2, 2, -5, 13), (0.43771272+0.98025439j))
((-4, -2, 0, -2, -4, -6, 8), (-0.31006136+1.24108134j))
((-4, -1, -2, -3, 0, 3, 18), (-0.3156383+1.23104938j))
((-4, 1, -2, 3, 0, -3, 18), (0.3156383+1.23104938j))
((-4, 2, 0, 2, -4, 6, 8), (0.31006136+1.24108134j))
((-1, 0, -1, -2, 3, 4, 59), (-0.12457457+1.96050879j))
((-1, 0, -1, 2, 3, -4, 59), (0.12457457+1.96050879j))
((0, -4, 0, 0, 0, -6, 0), 1.10668192j)
((0, -3, 0, 0, 0, -8, 0), 1.27788621j)
((0, -2, 0, 0, 0, -12, 0), 1.56508458j)
((0, -1, 0, 0, 0, -24, 0), 2.21336384j)
((0, 1, 0, 0, 0, 24, 0), 2.21336384j)
((0, 2, 0, 0, 0, 12, 0), 1.56508458j)
((0, 3, 0, 0, 0, 8, 0), 1.27788621j)
((0, 4, 0, 0, 0, 6, 0), 1.10668192j)
((1, 0, 1, -2, -3, 4, -59), (0.12457457+1.96050879j))
((1, 0, 1, 2, -3, -4, -59), (-0.12457457+1.96050879j))
((4, -2, 0, -2, 4, -6, -8), (0.31006136+1.24108134j))
((4, -1, 2, -3, 0, 3, -18), (0.3156383+1.23104938j))
((4, 1, 2, 3, 0, -3, -18), (-0.3156383+1.23104938j))
((4, 2, 0, 2, 4, 6, -8), (-0.31006136+1.24108134j))
((8, -4, 4, -2, -2, 5, -13), (0.43771272+0.98025439j))
((8, 4, 4, 2, -2, -5, -13), (-0.43771272+0.98025439j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-8, -4, -4, -2, 2, 5, 13) (-0.43771272+0.98025439j)
---- (8, 4, 4, 2, -2, -5, -13) (-0.43771272+0.98025439j)
Following forms are in same orbit
---- (-4, -2, 0, -2, -4, -6, 8) (-0.31006136+1.24108134j)
---- (4, 2, 0, 2, 4, 6, -8) (-0.31006136+1.24108134j)
Following forms are in same orbit
---- (-4, -1, -2, -3, 0, 3, 18) (-0.3156383+1.23104938j)
---- (4, 1, 2, 3, 0, -3, -18) (-0.3156383+1.23104938j)
```

```
Following forms are in same orbit
---- (-1, 0, -1, -2, 3, 4, 59) (-0.12457457+1.96050879j)
---- (1, 0, 1, 2, -3, -4, -59) (-0.12457457+1.96050879j)
Following forms are in same orbit
---- (0, -4, 0, 0, 0, -6, 0) 1.10668192j
---- (0, 4, 0, 0, 0, 6, 0) 1.10668192j
Following forms are in same orbit
---- (0, -3, 0, 0, 0, -8, 0) 1.27788621j
---- (0, 3, 0, 0, 0, 8, 0) 1.27788621j
Following forms are in same orbit
---- (0, -2, 0, 0, 0, -12, 0) 1.56508458j
---- (0, 2, 0, 0, 0, 12, 0) 1.56508458j
Following forms are in same orbit
---- (0, -1, 0, 0, 0, -24, 0) 2.21336384j
---- (0, 1, 0, 0, 0, 24, 0) 2.21336384j
step_4_results:
((-8, -4, -4, -2, 2, 5, 13), (-0.43771272+0.98025439j))
((-4, -1, -2, -3, 0, 3, 18), (-0.3156383+1.23104938j))
((-1, 0, -1, -2, 3, 4, 59), (-0.12457457+1.96050879j))
((0, -3, 0, 0, 0, -8, 0), 1.27788621j)
((0, -1, 0, 0, 0, -24, 0), 2.21336384j)
-----------------------------------------
r,d is  4 3
step_1_2_combined_results:
((-12, -10, -4, 0, 4, 10, 12), (-0.5+0.8660254j))
((-12, -2, 4, 6, 8, 8, 0), (-0.5+0.8660254j))
((-12, 2, 4, -6, 8, -8, 0), (0.5+0.8660254j))
((-12, 10, -4, 0, 4, -10, 12), (0.5+0.8660254j))
((-9, 0, 3, 0, 3, 0, -9), 1j)
((-3, -1, 2, 0, 4, 4, -24), 1.41421356j)
((-3, 1, 2, 0, 4, -4, -24), (-0+1.41421356j))
((-2, -6, -4, -2, 0, 8, 16), (-0.42602205+1.0601474j))
((-2, 0, 2, -2, 6, -4, -38), (0.34729636+1.62424005j))
((-2, 0, 2, 2, 6, 4, -38), (-0.34729636+1.62424005j))
((-2, 6, -4, 2, 0, -8, 16), (0.42602205+1.0601474j))
((0, -8, -8, -6, -4, 2, 12), (-0.5+0.8660254j))
((0, -8, 8, -6, 4, 2, -12), (0.5+0.8660254j))
((0, -6, 0, 0, 0, 6, 0), 1j)
((0, -4, 0, 0, 0, 9, 0), (-0+1.22474487j))
((0, -3, 0, 0, 0, 12, 0), 1.41421356j)
((0, -2, 0, 0, 0, 18, 0), (-0+1.73205081j))
((0, -1, 0, 0, 0, 36, 0), (-0+2.44948974j))
((0, 1, 0, 0, 0, -36, 0), (-0+2.44948974j))
((0, 2, 0, 0, 0, -18, 0), (-0+1.73205081j))
((0, 3, 0, 0, 0, -12, 0), 1.41421356j)
((0, 4, 0, 0, 0, -9, 0), (-0+1.22474487j))
((0, 6, 0, 0, 0, -6, 0), 1j)
((0, 8, -8, 6, -4, -2, 12), (0.5+0.8660254j))
((0, 8, 8, 6, 4, -2, -12), (-0.5+0.8660254j))
((2, -6, 4, -2, 0, 8, -16), (0.42602205+1.0601474j))
((2, 0, -2, -2, -6, -4, 38), (-0.34729636+1.62424005j))
((2, 0, -2, 2, -6, 4, 38), (0.34729636+1.62424005j))
((2, 6, 4, 2, 0, -8, -16), (-0.42602205+1.0601474j))
((3, -1, -2, 0, -4, 4, 24), (-0+1.41421356j))
((3, 1, -2, 0, -4, -4, 24), 1.41421356j)
((9, 0, -3, 0, -3, 0, 9), 1j)
((12, -10, 4, 0, -4, 10, -12), (0.5+0.8660254j))
```

```
((12, -2, -4, 6, -8, 8, 0), (0.5+0.8660254j))
((12, 2, -4, -6, -8, -8, 0), (-0.5+0.8660254j))
((12, 10, 4, 0, -4, -10, -12), (-0.5+0.8660254j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-12, -10, -4, 0, 4, 10, 12) (-0.5+0.8660254j)
---- (12, 10, 4, 0, -4, -10, -12) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-12, -2, 4, 6, 8, 8, 0) (-0.5+0.8660254j)
---- (0, -8, -8, -6, -4, 2, 12) (-0.5+0.8660254j)
---- (0, 8, 8, 6, 4, -2, -12) (-0.5+0.8660254j)
---- (12, 2, -4, -6, -8, -8, 0) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-9, 0, 3, 0, 3, 0, -9) 1j
---- (9, 0, -3, 0, -3, 0, 9) 1j
Following forms are in same orbit
---- (-3, -1, 2, 0, 4, 4, -24) 1.41421356j
---- (-3, 1, 2, 0, 4, -4, -24) (-0+1.41421356j)
---- (3, -1, -2, 0, -4, 4, 24) (-0+1.41421356j)
---- (3, 1, -2, 0, -4, -4, 24) 1.41421356j
Following forms are in same orbit
---- (-2, -6, -4, -2, 0, 8, 16) (-0.42602205+1.0601474j)
---- (2, 6, 4, 2, 0, -8, -16) (-0.42602205+1.0601474j)
Following forms are in same orbit
---- (-2, 0, 2, 2, 6, 4, -38) (-0.34729636+1.62424005j)
---- (2, 0, -2, -2, -6, -4, 38) (-0.34729636+1.62424005j)
Following forms are in same orbit
---- (0, -6, 0, 0, 0, 6, 0) 1j
---- (0, 6, 0, 0, 0, -6, 0) 1j
Following forms are in same orbit
---- (0, -4, 0, 0, 0, 9, 0) (-0+1.22474487j)
---- (0, 4, 0, 0, 0, -9, 0) (-0+1.22474487j)
Following forms are in same orbit
---- (0, -3, 0, 0, 0, 12, 0) 1.41421356j
---- (0, 3, 0, 0, 0, -12, 0) 1.41421356j
Following forms are in same orbit
---- (0, -2, 0, 0, 0, 18, 0) (-0+1.73205081j)
---- (0, 2, 0, 0, 0, -18, 0) (-0+1.73205081j)
Following forms are in same orbit
---- (0, -1, 0, 0, 0, 36, 0) (-0+2.44948974j)
---- (0, 1, 0, 0, 0, -36, 0) (-0+2.44948974j)
step_4_results:
((-3, -1, 2, 0, 4, 4, -24), 1.41421356j)
((0, -4, 0, 0, 0, 9, 0), (-0+1.22474487j))
((0, -1, 0, 0, 0, 36, 0), (-0+2.44948974j))
------------------------------------------
r,d is  4 -3
step_1_2_combined_results:
((-15, -8, -5, 0, 5, 8, 15), (-0.5+0.8660254j))
((-15, -7, -4, -6, -8, -8, 0), (-0.5+0.8660254j))
((-15, 7, -4, 6, -8, 8, 0), (0.5+0.8660254j))
((-15, 8, -5, 0, 5, -8, 15), (0.5+0.8660254j))
((-9, 0, -3, 0, 3, 0, 9), 1j)
((-8, -4, 0, -2, -4, -6, 4), (-0.26099353+1.03869149j))
((-8, -3, -4, -4, 0, 4, 16), (-0.38492456+1.04018459j))
((-8, 3, -4, 4, 0, -4, 16), (0.38492456+1.04018459j))
```

```
((-8, 4, 0, 2, -4, 6, 4), (0.26099353+1.03869149j))
((-2, -1, 0, -3, -6, -9, 36), (-0.37418318+1.69114959j))
((-2, 1, 0, 3, -6, 9, 36), (0.37418318+1.69114959j))
((-1, 0, 1, -4, 3, -8, 101), (0.47801294+2.07738298j))
((-1, 0, 1, 4, 3, 8, 101), (-0.47801294+2.07738298j))
((0, -8, -8, -6, -4, -7, -15), (-0.5+0.8660254j))
((0, -8, 8, -6, 4, -7, 15), (0.5+0.8660254j))
((0, -6, 0, 0, 0, -6, 0), 1j)
((0, -4, 0, 0, 0, -9, 0), 1.22474487j)
((0, -3, 0, 0, 0, -12, 0), (-0+1.41421356j))
((0, -2, 0, 0, 0, -18, 0), 1.73205081j)
((0, -1, 0, 0, 0, -36, 0), 2.44948974j)
((0, 1, 0, 0, 0, 36, 0), 2.44948974j)
((0, 2, 0, 0, 0, 18, 0), 1.73205081j)
((0, 3, 0, 0, 0, 12, 0), (-0+1.41421356j))
((0, 4, 0, 0, 0, 9, 0), 1.22474487j)
((0, 6, 0, 0, 0, 6, 0), 1j)
((0, 8, -8, 6, -4, 7, -15), (0.5+0.8660254j))
((0, 8, 8, 6, 4, 7, 15), (-0.5+0.8660254j))
((1, 0, -1, -4, -3, -8, -101), (-0.47801294+2.07738298j))
((1, 0, -1, 4, -3, 8, -101), (0.47801294+2.07738298j))
((2, -1, 0, -3, 6, -9, -36), (0.37418318+1.69114959j))
((2, 1, 0, 3, 6, 9, -36), (-0.37418318+1.69114959j))
((8, -4, 0, -2, 4, -6, -4), (0.26099353+1.03869149j))
((8, -3, 4, -4, 0, 4, -16), (0.38492456+1.04018459j))
((8, 3, 4, 4, 0, -4, -16), (-0.38492456+1.04018459j))
((8, 4, 0, 2, 4, 6, -4), (-0.26099353+1.03869149j))
((9, 0, 3, 0, -3, 0, -9), 1j)
((15, -8, 5, 0, -5, 8, -15), (0.5+0.8660254j))
((15, -7, 4, -6, 8, -8, 0), (0.5+0.8660254j))
((15, 7, 4, 6, 8, 8, 0), (-0.5+0.8660254j))
((15, 8, 5, 0, -5, -8, -15), (-0.5+0.8660254j))
step_3_results:
here are the GL2Z equivalent classes
Following forms are in same orbit
---- (-15, -8, -5, 0, 5, 8, 15) (-0.5+0.8660254j)
---- (15, 8, 5, 0, -5, -8, -15) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-15, -7, -4, -6, -8, -8, 0) (-0.5+0.8660254j)
---- (0, -8, -8, -6, -4, -7, -15) (-0.5+0.8660254j)
---- (0, 8, 8, 6, 4, 7, 15) (-0.5+0.8660254j)
---- (15, 7, 4, 6, 8, 8, 0) (-0.5+0.8660254j)
Following forms are in same orbit
---- (-9, 0, -3, 0, 3, 0, 9) 1j
---- (9, 0, 3, 0, -3, 0, -9) 1j
Following forms are in same orbit
---- (-8, -4, 0, -2, -4, -6, 4) (-0.26099353+1.03869149j)
---- (8, 4, 0, 2, 4, 6, -4) (-0.26099353+1.03869149j)
Following forms are in same orbit
---- (-8, -3, -4, -4, 0, 4, 16) (-0.38492456+1.04018459j)
---- (8, 3, 4, 4, 0, -4, -16) (-0.38492456+1.04018459j)
Following forms are in same orbit
---- (-2, -1, 0, -3, -6, -9, 36) (-0.37418318+1.69114959j)
---- (2, 1, 0, 3, 6, 9, -36) (-0.37418318+1.69114959j)
Following forms are in same orbit
---- (-1, 0, 1, 4, 3, 8, 101) (-0.47801294+2.07738298j)
---- (1, 0, -1, -4, -3, -8, -101) (-0.47801294+2.07738298j)
```

```
Following forms are in same orbit
---- (0, -6, 0, 0, 0, -6, 0) 1j
---- (0, 6, 0, 0, 0, 6, 0) 1j
Following forms are in same orbit
---- (0, -4, 0, 0, 0, -9, 0) 1.22474487j
---- (0, 4, 0, 0, 0, 9, 0) 1.22474487j
Following forms are in same orbit
---- (0, -3, 0, 0, 0, -12, 0) (-0+1.41421356j)
---- (0, 3, 0, 0, 0, 12, 0) (-0+1.41421356j)
Following forms are in same orbit
---- (0, -2, 0, 0, 0, -18, 0) 1.73205081j
---- (0, 2, 0, 0, 0, 18, 0) 1.73205081j
Following forms are in same orbit
---- (0, -1, 0, 0, 0, -36, 0) 2.44948974j
---- (0, 1, 0, 0, 0, 36, 0) 2.44948974j
step_4_results:
((-15, -8, -5, 0, 5, 8, 15), (-0.5+0.8660254j))
((-15, -7, -4, -6, -8, -8, 0), (-0.5+0.8660254j))
((-8, -3, -4, -4, 0, 4, 16), (-0.38492456+1.04018459j))
((-2, -1, 0, -3, -6, -9, 36), (-0.37418318+1.69114959j))
((-1, 0, 1, 4, 3, 8, 101), (-0.47801294+2.07738298j))
((0, -4, 0, 0, 0, -9, 0), 1.22474487j)
((0, -1, 0, 0, 0, -36, 0), 2.44948974j)
```

Following are the final results

In [39]:

```
for (r,d), result in zip(parameters_to_run, results):
    print('----------------------------------')
    print("r,d = ",r,d)
    for form in result:
        print(form)
```

```
----------------------------------
r,d =  3 1
((-2, -1, 0, -1, -2), (-0.26794919+0.96343304j))
((-1, 0, -1, 0, 3), (-0+1.31607401j))
((-1, 0, 0, -2, 0), 1.41421356j)
((-1, 1, 1, 1, -1), (-0.26794919+0.96343304j))
((0, -1, 0, 0, -4), 1.41421356j)
((0, -1, 2, -3, 0), 1.41421356j)
((1, 0, -1, 0, -3), (-0+1.31607401j))
----------------------------------
r,d =  3 -1
((-1, 0, 1, 0, 3), (-0+1.31607401j))
((0, -1, 0, 0, 4), 1.41421356j)
((0, 1, -2, 3, 0), 1.41421356j)
((1, -1, -1, -1, 1), (-0.26794919+0.96343304j))
((1, 0, 0, -2, 0), 1.41421356j)
((1, 0, 1, 0, -3), (-0+1.31607401j))
((2, 1, 0, 1, 2), (-0.26794919+0.96343304j))
----------------------------------
r,d =  3 2
((-1, 0, -1, -2, 3), (-0.15826276+1.50897958j))
((0, -1, 0, 0, -8), 1.78179744j)
((1, 2, 1, 0, -3), (-0.43612414+1.0108584j))
----------------------------------
r,d =  3 -2
((-1, -2, -1, 0, 3), (-0.43612414+1.0108584j))
((0, -1, 0, 0, 8), 1.78179744j)
((1, 0, 1, 2, -3), (-0.15826276+1.50897958j))
----------------------------------
r,d =  3 3
((-2, -1, 0, -2, -4), (-0.31822819+1.18033017j))
((-1, -1, 0, -2, -8), (-0.42588044+1.57961975j))
((0, -2, 0, 0, -3), 1.01982445j)
((0, -1, 0, 0, -12), 2.0396489j)
----------------------------------
r,d =  3 -3
((0, -2, 0, 0, 3), 1.01982445j)
((0, -1, 0, 0, 12), 2.0396489j)
((1, 1, 0, 2, 8), (-0.42588044+1.57961975j))
((2, 1, 0, 2, 4), (-0.31822819+1.18033017j))
----------------------------------
r,d =  3 4
((-3, -1, -1, -3, -3), (-0.43612414+1.0108584j))
((-3, 0, -1, -2, 1), (-0.11501333+0.99336395j))
((-1, 0, 0, -4, 0), 1.78179744j)
((0, -1, 0, 0, -16), 2.2449241j)
((1, 1, -1, -1, -7), (-0.15826276+1.50897958j))
----------------------------------
r,d =  3 -4
((-1, -1, 1, 1, 7), (-0.15826276+1.50897958j))
((-1, 2, 1, 0, 3), (-0.11501333+0.99336395j))
((0, -1, 0, 0, 16), 2.2449241j)
((1, 0, 0, -4, 0), 1.78179744j)
((3, 1, 1, 3, 3), (-0.43612414+1.0108584j))
----------------------------------
r,d =  4 1
((-3, -4, -1, 0, 1, 4, 3), (-0.26794919+0.96343304j))
```

```
((-1, 0, 1, 0, 3, 0, -27), (-0+1.73205081j))
((0, -3, 0, 0, 0, 4, 0), 1.07456993j)
((0, -1, 0, 0, 0, 12, 0), 1.86120972j)
----------------------------------
r,d =  4 -1
((-8, -4, -4, -4, -2, 1, 7), (-0.5+0.8660254j))
((-7, -6, -3, -2, -3, -6, -7), (-0.5+0.8660254j))
((-5, -1, 1, 3, 3, 3, 9), (-0.43612414+1.0108584j))
((-1, 0, -1, 0, 3, 0, 27), 1.73205081j)
((-1, 0, 0, -2, 0, 0, 32), (-0+1.78179744j))
((-1, 1, 1, 1, -1, 5, 17), (-0.15826276+1.50897958j))
((0, -3, 0, 0, 0, -4, 0), 1.07456993j)
((0, -1, 0, 0, 0, -12, 0), (-0+1.86120972j))
----------------------------------
r,d =  4 2
((-5, -1, 2, 2, 4, 4, -8), (-0.31010205+1.13297512j))
((-1, 1, 2, 2, 4, -4, -40), (-0.44948974+1.64223581j))
((0, -3, 0, 0, 0, 8, 0), (-0+1.27788621j))
((0, -1, 0, 0, 0, 24, 0), 2.21336384j)
----------------------------------
r,d =  4 -2
((-8, -4, -4, -2, 2, 5, 13), (-0.43771272+0.98025439j))
((-4, -1, -2, -3, 0, 3, 18), (-0.3156383+1.23104938j))
((-1, 0, -1, -2, 3, 4, 59), (-0.12457457+1.96050879j))
((0, -3, 0, 0, 0, -8, 0), 1.27788621j)
((0, -1, 0, 0, 0, -24, 0), 2.21336384j)
----------------------------------
r,d =  4 3
((-3, -1, 2, 0, 4, 4, -24), 1.41421356j)
((0, -4, 0, 0, 0, 9, 0), (-0+1.22474487j))
((0, -1, 0, 0, 0, 36, 0), (-0+2.44948974j))
----------------------------------
r,d =  4 -3
((-15, -8, -5, 0, 5, 8, 15), (-0.5+0.8660254j))
((-15, -7, -4, -6, -8, -8, 0), (-0.5+0.8660254j))
((-8, -3, -4, -4, 0, 4, 16), (-0.38492456+1.04018459j))
((-2, -1, 0, -3, -6, -9, 36), (-0.37418318+1.69114959j))
((-1, 0, 1, 4, 3, 8, 101), (-0.47801294+2.07738298j))
((0, -4, 0, 0, 0, -9, 0), 1.22474487j)
((0, -1, 0, 0, 0, -36, 0), 2.44948974j)
```

In [ ]:

In [ ]:

In [ ]: