CODE:

```
package logdelpezzo;

/**
 *
 * @author justinlacini
 */
public class Fraction {

    private int n,d;

    public Fraction(int numerator, int denominator){
        this.setFraction(numerator, denominator);
    }

    public Fraction(){
        n=0;
        d=1;
    }


    public void setFraction(int numerator, int denominator) {
        n=numerator;
        d=denominator;
        this.simplify();
    }

    public int getNumerator(){
        return n;
    }

    public int getDenominator(){
        return d;
    }

    private void simplify(){
        if (d==0) {
            n=0;
            return;
        }
        else{
            int gcd = gcd(n,d);
            n = n / gcd;
            d = d / gcd;
            if (n==0)
                d=1;
        }
    }

    private int gcd(int a, int b){
        int pa = Math.abs(a); //positive a
        int pb = Math.abs(b); //positive b
        int t;
```

```
        while(pb != 0){
            t = pb;
            pb = pa % pb;
            pa = t;
        }
        return pa;
    }
}

package logdelpezzo;

/**
 *
 * @author justinlacini
 */
public class Operation {

    public Fraction add(Fraction f, Fraction g){
        int d = f.getDenominator() * g.getDenominator();
        int n = f.getNumerator() * g.getDenominator() + f.getDenominator() * g.getNumerator();
        Fraction addition = new Fraction(n, d);
        return addition;
    }

    public Fraction add(Fraction f, int a) {
        Fraction fa = new Fraction(a,1);
        return this.add(f,fa);
    }

    public Fraction times(Fraction f, Fraction g){
        Fraction multiplication = new Fraction();
        int n = f.getNumerator()*g.getNumerator();
        int d = f.getDenominator()*g.getDenominator();
        multiplication.setFraction(n, d);
        return multiplication;
    }

    public boolean equal(Fraction f, Fraction g) {
        int a = f.getNumerator()*g.getDenominator();
        int b = f.getDenominator()*g.getNumerator();
        return (a==b);
    }

    public boolean equal(Fraction f, int a) {
        return (a*f.getDenominator() == f.getNumerator());
    }

    public Fraction minus(Fraction f){
        Fraction minus = new Fraction();
        minus.setFraction(-f.getNumerator(), f.getDenominator());
        return minus;
    }

    public Fraction minus(Fraction f, Fraction g){
```

```
        return this.add(f, this.minus(g));
    }

}

package logdelpezzo;

/**
 *
 * @author justinlacini
 */
public class Main {

    public static void main(String[] args) {



        System.out.println("config I");

        int g,R,v; int k,j;
        g=v=j=k=R=0;
        int[] n = new int[9];

        Fraction f = new Fraction();
        Fraction h = new Fraction();
        Fraction s = new Fraction();
        Fraction p = new Fraction();

        Operation op = new Operation();

        for(R = 0; R < 6; R++)
        for(v = 0; v < 10; v++)
        for(g = 1; g<=v+1; g++)
        for(n[0]=0; n[0]+v <=15; n[0]++)
        for(n[1]=0; n[1]+v <= 15; n[1]++)
        for(n[2]=0; n[1]+2*n[2]+v <= 15; n[2]++)
        for(n[3]=0; n[1]+2*n[2]+3*n[3]+v <= 15; n[3]++)
        for(n[4]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+v <= 15; n[4]++)
        for(n[5]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+v <= 15; n[5]++)
        for(n[6]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+v <= 15; n[6]++)
        for(n[7]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+v <= 15; n[7]++){
            k= (2*R + 3)*(4*g*R+4*g-2*R-3);
            f.setFraction(g, k);
            h= new Fraction();
            s= new Fraction();
            for(int t=0; t < 8; t++){
                p.setFraction(2*n[t]*(t+1)*(t+1), 2*t+3);
                h=op.add(h, p);
            }
            s=op.add(f,h);
            s=op.add(s,v);
            if((op.equal(s, 9))&& v!=9) {
                System.out.println("g = "+g+" R= "+R+" v= "+v+" n0 = "+n[0]+
                        " n1 = "+n[1]+" n2 = "+n[2]+" n3 = "+n[3]+" n4 = "+n[4]+
```

```
            " n5 = "+n[5]+" n6 = "+n[6]+" n7 = "+n[7]+" n8 = "+n[8]);
    }
}



System.out.println("configuration II, case x_0 = (3, A_g)");


g=v=j=k=0;
n = new int[9];

f = new Fraction();
h = new Fraction();
s = new Fraction();
p = new Fraction();

op = new Operation();

for(v = 0; v < 8; v++)
for(g = 2; g<=8; g++)
for(n[0]=0; n[0]+v <=15; n[0]++)
for(n[1]=0; n[1]+v <= 15; n[1]++)
for(n[2]=0; n[1]+2*n[2]+v <= 15; n[2]++)
for(n[3]=0; n[1]+2*n[2]+3*n[3]+v <= 15; n[3]++)
for(n[4]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+v <= 15; n[4]++)
for(n[5]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+v <= 15; n[5]++)
for(n[6]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+v <= 15; n[6]++)
for(n[7]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+v <= 15; n[7]++){
//for(n[8]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+8*n[8]+v <= 9 ; n[8]++) {
    //System.out.println("working");
    k= (2*g + 3)*(2*g - 1);
    f.setFraction(2, k);
    h= new Fraction();
    s= new Fraction();
    for(int t=0; t < 9; t++){
        p.setFraction(2*n[t]*(t+1)*(t+1), 2*t+3);
        h=op.add(h, p);
    }
    s=op.add(f,h);
    s=op.add(s,v);
    //if(v==8) System.out.println(s.getNumerator()+"/"+s.getDenominator());
    if((op.equal(s, 9))&& v!=9) {
        System.out.println("g = "+g+" v= "+v+" n0 = "+n[0]+
                " n1 = "+n[1]+" n2 = "+n[2]+" n3 = "+n[3]+" n4 = "+n[4]+
                " n5 = "+n[5]+" n6 = "+n[6]+" n7 = "+n[7]+" n8 = "+n[8]);
    }
}



System.out.println("config III case x_0 = (4)");
```

```
g=v=j=k=0;
n = new int[9];

f = new Fraction();
h = new Fraction();
s = new Fraction();
p = new Fraction();

op = new Operation();

for(v = 1; v < 9; v++)
for(g = 2; g<=5; g++)
for(n[0]=0; n[0]+v <=15; n[0]++)
for(n[1]=0; n[1]+v <= 15; n[1]++)
for(n[2]=0; n[1]+2*n[2]+v <= 15; n[2]++)
for(n[3]=0; n[1]+2*n[2]+3*n[3]+v <= 15; n[3]++)
for(n[4]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+v <= 15; n[4]++)
for(n[5]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+v <= 15; n[5]++)
for(n[6]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+v <= 15; n[6]++)
for(n[7]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+v <= 15; n[7]++){
//for(n[8]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+8*n[8]+v <= 9 ; n[8]++) {
    //System.out.println("working");
    k= (2*g + 1)*(2*g - 1);
    f.setFraction(1, k);
    h= new Fraction();
    s= new Fraction();
    for(int t=0; t < 9; t++){
        p.setFraction(2*n[t]*(t+1)*(t+1), 2*t+3);
        h=op.add(h, p);
    }
    s=op.add(f,h);
    s=op.add(s,v);
    //if(v==8) System.out.println(s.getNumerator()+"/"+s.getDenominator());
    if((op.equal(s, 9))&& v!=9) {
        System.out.println("g = "+g+" v= "+v+" n0 = "+n[0]+
                " n1 = "+n[1]+" n2 = "+n[2]+" n3 = "+n[3]+" n4 = "+n[4]+
                " n5 = "+n[5]+" n6 = "+n[6]+" n7 = "+n[7]+" n8 = "+n[8]);
    }
}




System.out.println("config III case x_0 = (3, A_1, 3) ");

//change A_k here based on the formula on my notes

g=v=j=k=0;
n = new int[9];

f = new Fraction();
h = new Fraction();
s = new Fraction();
p = new Fraction();
```

```
op = new Operation();

for(v = 1; v < 9; v++)
for(g = 2; g<=6; g++)
for(n[0]=0; n[0]+v <=15; n[0]++)
for(n[1]=0; n[1]+v <= 15; n[1]++)
for(n[2]=0; n[1]+2*n[2]+v <= 15; n[2]++)
for(n[3]=0; n[1]+2*n[2]+3*n[3]+v <= 15; n[3]++)
for(n[4]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+v <= 15; n[4]++)
for(n[5]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+v <= 15; n[5]++)
for(n[6]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+v <= 15; n[6]++)
for(n[7]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+v <= 15; n[7]++){
//for(n[8]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+8*n[8]+v <= 9 ; n[8]++) {
    //System.out.println("working");
    k= (2*g + 1)*(10*g - 1);
    f.setFraction(3, k);
    h= new Fraction();
    s= new Fraction();
    for(int t=0; t < 9; t++){
        p.setFraction(2*n[t]*(t+1)*(t+1), 2*t+3);
        h=op.add(h, p);
    }
    s=op.add(f,h);
    s=op.add(s,v);
    //if(v==8) System.out.println(s.getNumerator()+"/"+s.getDenominator());
    if((op.equal(s, 9))&& v!=9) {
        System.out.println("g = "+g+" v= "+v+" n0 = "+n[0]+
                " n1 = "+n[1]+" n2 = "+n[2]+" n3 = "+n[3]+" n4 = "+n[4]+
                " n5 = "+n[5]+" n6 = "+n[6]+" n7 = "+n[7]+" n8 = "+n[8]);
    }
}




System.out.println("config III e<1/2");

g=v=j=k=R=0;
n = new int[9];

f = new Fraction();
h = new Fraction();
s = new Fraction();
p = new Fraction();

op = new Operation();

for(R = 1; R < 5; R++)
for(v = 1; v < 7; v++)
for(g = 2; g<=3 && g-1<=R; g++)
for(n[0]=0; n[0]+v <=15; n[0]++)
for(n[1]=0; n[1]+v <= 15; n[1]++)
for(n[2]=0; n[1]+2*n[2]+v <= 15; n[2]++)
```

```
        for(n[3]=0; n[1]+2*n[2]+3*n[3]+v <= 15; n[3]++)
        for(n[4]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+v <= 15; n[4]++)
        for(n[5]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+v <= 15; n[5]++)
        for(n[6]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+v <= 15; n[6]++)
        for(n[7]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+v <= 15; n[7]++){
        //for(n[8]=0; n[1]+2*n[2]+3*n[3]+4*n[4]+5*n[5]+6*n[6]+7*n[7]+8*n[8]+v <= 9 ; n[8]++) {
            //System.out.println("working");
            k= (2*g+1)*(4*g+4*2*R-1)*(2*R+3);
            j = 2*(g+R+2)*(g+R+2);
            f.setFraction(j, k);
            h= new Fraction();
            s= new Fraction();
            for(int t=0; t < 9; t++){
                p.setFraction(2*n[t]*(t+1)*(t+1), 2*t+3);
                h=op.add(h, p);
            }
            s=op.add(f,h);
            s=op.add(s,v);
            //if(v==8) System.out.println(s.getNumerator()+"/"+s.getDenominator());
            if((op.equal(s, 9))&& v!=9) {
                System.out.println("g = "+g+" R= "+R+" v= "+v+" n0 = "+n[0]+
                        " n1 = "+n[1]+" n2 = "+n[2]+" n3 = "+n[3]+" n4 = "+n[4]+
                        " n5 = "+n[5]+" n6 = "+n[6]+" n7 = "+n[7]+" n8 = "+n[8]);
            }
        }



    }

}
```

PRINTOUT:

config I
g = 1 R= 0 v= 0 n0 = 1 n1 = 5 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 0 v= 0 n0 = 13 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 R= 0 v= 1 n0 = 2 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 1 n7 = 0 n8 = 0
g = 2 R= 0 v= 1 n0 = 7 n1 = 2 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 0 v= 2 n0 = 10 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 3 R= 0 v= 2 n0 = 5 n1 = 0 n2 = 0 n3 = 1 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 R= 0 v= 3 n0 = 4 n1 = 2 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 0 v= 4 n0 = 7 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 3 R= 0 v= 4 n0 = 2 n1 = 0 n2 = 0 n3 = 1 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 R= 0 v= 5 n0 = 1 n1 = 2 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 6 R= 0 v= 5 n0 = 2 n1 = 0 n2 = 1 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 0 v= 6 n0 = 4 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 0 v= 8 n0 = 1 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 1 v= 0 n0 = 11 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 1 v= 2 n0 = 8 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 1 v= 4 n0 = 5 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 1 v= 6 n0 = 2 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 1 R= 2 v= 0 n0 = 0 n1 = 4 n2 = 1 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
configuration II, case x_0 = (3, A_g)
g = 2 v= 1 n0 = 8 n1 = 0 n2 = 1 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 v= 3 n0 = 5 n1 = 0 n2 = 1 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 v= 5 n0 = 2 n1 = 0 n2 = 1 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
config III case x_0 = (4)
g = 2 v= 2 n0 = 8 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 v= 4 n0 = 5 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 v= 6 n0 = 2 n1 = 1 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
config III case x_0 = (3, A_1, 3)
config III e<1/2
g = 2 R= 1 v= 1 n0 = 2 n1 = 0 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 1 n7 = 0 n8 = 0
g = 2 R= 1 v= 1 n0 = 7 n1 = 2 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 R= 1 v= 3 n0 = 4 n1 = 2 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0
g = 2 R= 1 v= 5 n0 = 1 n1 = 2 n2 = 0 n3 = 0 n4 = 0 n5 = 0 n6 = 0 n7 = 0 n8 = 0