```
/* 2次関数ラーニング */

import java.io.*;

class calcY
{
	private double a;
	private double b;

	public double getA(){
		return a;
	}
	public double getB(){
		return b;
	}

	public void calc(double x1, double y1, double x2, double y2) throws IOException
	{
		/* y1 = a * x1 + b */
		/* y2 = a * x2 + b */

		a = 0;
		b = 0;

		if((x1) == 0){
			b = (y1);
			if((x2) != 0){
				a = ((y2) - (b)) / (x2);
```

```
            }else{
                throw new IOException();
            }
        }else{
            /* y1 = a * x1 + b */
            /* y2 = a * x2 + b */
            /* a * x1 - y1 = a * x2 - y2 */
            /* a * (x1 - x2) = y1 - y2 */

            if((x1) != (x2)){
                a = ((y1) - (y2)) / ((x1) - (x2));
                b = (y1) - ((a) * (x1));
            }else{
                throw new IOException();
            }
        }

        return;
    }
}

class calcYX
{
    private double a;
    private double b;
    private double c;

    public double getA(){
        return a;
```

```
    }
    public double getB(){
        return b;
    }
    public double getC(){
        return c;
    }

    public void calc(double x1, double y1, double x2, double y2, double x3, double y3) throws
IOException
    {
        /*
            y1 = a * x1 * x1 + b * x1 + c = (a * x1 + b) * x1 + c = yx1 * x1 + c
            y2 = a * x2 * x2 + b * x2 + c = (a * x2 + b) * x2 + c = yx2 * x2 + c
            y3 = a * x3 * x3 + b * x3 + c = (a * x3 + b) * x3 + c = yx3 * x3 + c
            c = y1-a*x1*x1-b*x1 = y2-a*x2*x2-b*x2
            (y1-y2)+a*((x2*x2)-(x1*x1))+b(x2-x1) = 0
            ((y1-y2)/(x2-x1))+a*(x2+x1)+b = 0
            b = ((y1-y2)/(x1-x2))-a*(x1+x2) = ((y2-y3)/(x2-x3))-a*(x2+x3)
            a*(x3-x1) = (((y2-y3)*(x1-x2))-((y1-y2)*(x2-x3)))/((x1-x2)*(x2-x3))
            a = (x1*y2-x1*y3-x2*y1+x2*y3+x3*y1-x3*y2)/((x1-x2)*(x2-x3)*(x3-x1))
            yx1 = a*x1+b = a*x1+((y1-y2)/(x1-x2))-a*(x1+x2)=((y1-y2)/(x1-x2))-a*x2
              = ((y1-y2)/(x1-x2))-((x2*(x1*y2-x1*y3-x2*y1+x2*y3+x3*y1-x3*y2))/((x1-x2)*(x2-
x3)*(x3-x1)))
            c = y1-yx1*x1
        */

        double yx1 = ((((y1)-(y2))/((x1)-(x2)))-(((x2)*(((x1)*(y2))-((x1)*(y3))-
((x2)*(y1))+((x2)*(y3))+((x3)*(y1))-((x3)*(y2))))/(((x1)-(x2))*((x2)-(x3))*((x3)-(x1)))));
```

```
        double yx2 = ((((y2)-(y3))/((x2)-(x3)))-(((x3)*(((x2)*(y3))-((x2)*(y1))-
((x3)*(y2))+((x3)*(y1))+((x1)*(y2))-((x1)*(y3))))/(((x2)-(x3))*((x3)-(x1))*((x1)-(x2)))));
        double yx3 = ((((y3)-(y1))/((x3)-(x1)))-(((x1)*(((x3)*(y1))-((x3)*(y2))-
((x1)*(y3))+((x1)*(y2))+((x2)*(y3))-((x2)*(y1))))/(((x3)-(x1))*((x1)-(x2))*((x2)-(x3)))));

        calcY cyb = new calcY();
        cyb.calc(x1, yx1, x2, yx2);
        a = cyb.getA();
        b = cyb.getB();
        c = ((y1)-((yx1)*(x1)));

        return;
    }
}

public class learning20181229
{
    public static void main(String args[])
    {
        try{
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            String buf = null;

            int l = 1;
            int m = 3;
            int n = l * m + 1;

            double[] x1 = new double[n];
```

```
double[] x2 = new double[n];
double[] x3 = new double[n];
double[] x4 = new double[n];
double[] y1 = new double[n];
double[] y2 = new double[n];
double[] y3 = new double[n];
double[] z1 = new double[n];
double[] z2 = new double[n];
double[] z3 = new double[n];
double[] z4 = new double[n];
double[] a = new double[n];
double[] b = new double[n];
double[] c = new double[n];

for(int i = 0; i < n; i++){
    x1[i] = 0;
    x2[i] = 0;
    x3[i] = 0;
    x4[i] = 0;
    y1[i] = 0;
    y2[i] = 0;
    y3[i] = 0;
    z1[i] = 0;
    z2[i] = 0;
    z3[i] = 0;
    z4[i] = 0;
    a[i] = 0;
    b[i] = 0;
```

```
        c[i] = 0;
    }

    for(int i = 0; i < n; i++){
        System.out.print("要素" + i + "入力1:");
        buf = br.readLine();
        x1[i] = Double.parseDouble(buf);
        System.out.print("測定値" + i + "入力1:");
        buf = br.readLine();
        z1[i] = Double.parseDouble(buf);
    }

    for(int i = 0; i < n; i++){
        System.out.print("要素" + i + "入力2:");
        buf = br.readLine();
        x2[i] = Double.parseDouble(buf);
        System.out.print("測定値" + i + "入力2:");
        buf = br.readLine();
        z2[i] = Double.parseDouble(buf);
    }

    for(int i = 0; i < n; i++){
        System.out.print("要素" + i + "入力3:");
        buf = br.readLine();
        x3[i] = Double.parseDouble(buf);
        System.out.print("測定値" + i + "入力3:");
        buf = br.readLine();
        z3[i] = Double.parseDouble(buf);
    }
```

```
            /* z1[0] = y1[0] + y1[1] + y1[2] = a[0] * x1[0] + a[1] * x1[1] + a[2] * x1[2] + b[0] + b[1] +
b[2] */
            /* z2[0] = y2[0] + y2[1] + y2[2] = a[0] * x2[0] + a[1] * x2[1] + a[2] * x2[2] + b[0] + b[1] +
b[2] */
            /* z3[0] = y3[0] + y3[1] + y3[2] = a[0] * x3[0] + a[1] * x3[1] + a[2] * x3[2] + b[0] + b[1] +
b[2] */

            /* z1[0] = y1[0] + y1[1] + y1[2] */
            /* Z1 = 3 * Y1 */
            /* Y1 = y1[0] + z1[1] + z1[4] + z1[7] */
            /* y1[0] = (Z1 / 3) - z1[1] - z1[4] - z1[7] */

            double Z1 = 0;
            double Z2 = 0;
            double Z3 = 0;

            for(int i = 0; i < n; i++){
                Z1 += (z1[i]);
                Z2 += (z2[i]);
                Z3 += (z3[i]);
            }

            for(int i = 0; i < n; i++){
                y1[i] = ((Z1) / m);
                y2[i] = ((Z2) / m);
                y3[i] = ((Z3) / m);
                for(int k = 0; k < l; k++){
                    y1[i] -= z1[(i + 1 + (m * k)) % n];
```

```
                    y2[i] -= z2[(i + 1 + (m * k)) % n];
                    y3[i] -= z3[(i + 1 + (m * k)) % n];
                }
                calcYX cyx = new calcYX();
                cyx.calc(x1[i], y1[i], x2[i], y2[i], x3[i], y3[i]);
                a[i] = cyx.getA();
                b[i] = cyx.getB();
                c[i] = cyx.getC();
                y1[i] = (a[i]) * (x1[i]) * (x1[i]) + (b[i]) * (x1[i]) + (c[i]);
                y2[i] = (a[i]) * (x2[i]) * (x2[i]) + (b[i]) * (x2[i]) + (c[i]);
                y3[i] = (a[i]) * (x3[i]) * (x3[i]) + (b[i]) * (x3[i]) + (c[i]);
                System.out.println("y[" + i + "] = " + (a[i]) + "x[" + i + "]x[" + i + "] + " + (b[i]) + "x[" +
i + "] + " + (c[i]));
            }

            for(int i = 0; i < n; i++){
                z1[i] = 0;
                z2[i] = 0;
                for(int j = 0; j < m; j++){
                    z1[i] += (y1[(i + j) % n]);
                    z2[i] += (y2[(i + j) % n]);
                }
                System.out.print("z[" + i + "] = ");
                for(int j = 0; j < m; j++){
                    System.out.print((a[(i + j) % n]) + "x[" + ((i + j) % n) + "]x[" + ((i + j) % n) + "] +
");
                    System.out.print((b[(i + j) % n]) + "x[" + ((i + j) % n) + "] + ");
                }
                double C = 0;
```

```
                for(int j = 0; j < m; j++){
                    C += (c[(i + j) % n]);
                }
                System.out.println(C);
            }

            for(int i = 0; i < n; i++){
                System.out.print("要素" + i + "入力4:");
                buf = br.readLine();
                x4[i] = Double.parseDouble(buf);
                z4[i] = 0;
            }

            for(int i = 0; i < n; i++){
                for(int j = 0; j < m; j++){
                    z4[i] += ((a[(i + j) % n]) * (x4[(i + j) % n]) * (x4[(i + j) % n]) + (b[(i + j) % n]) *
(x4[(i + j) % n]) + (c[(i + j) % n]));
                }
                System.out.println("予測" + i + "出力4 = " + (z4[i]));
            }

        }catch(IOException e){
            System.out.println("例外" + e + "が発生しました");
        }

        return;
    }
}
```