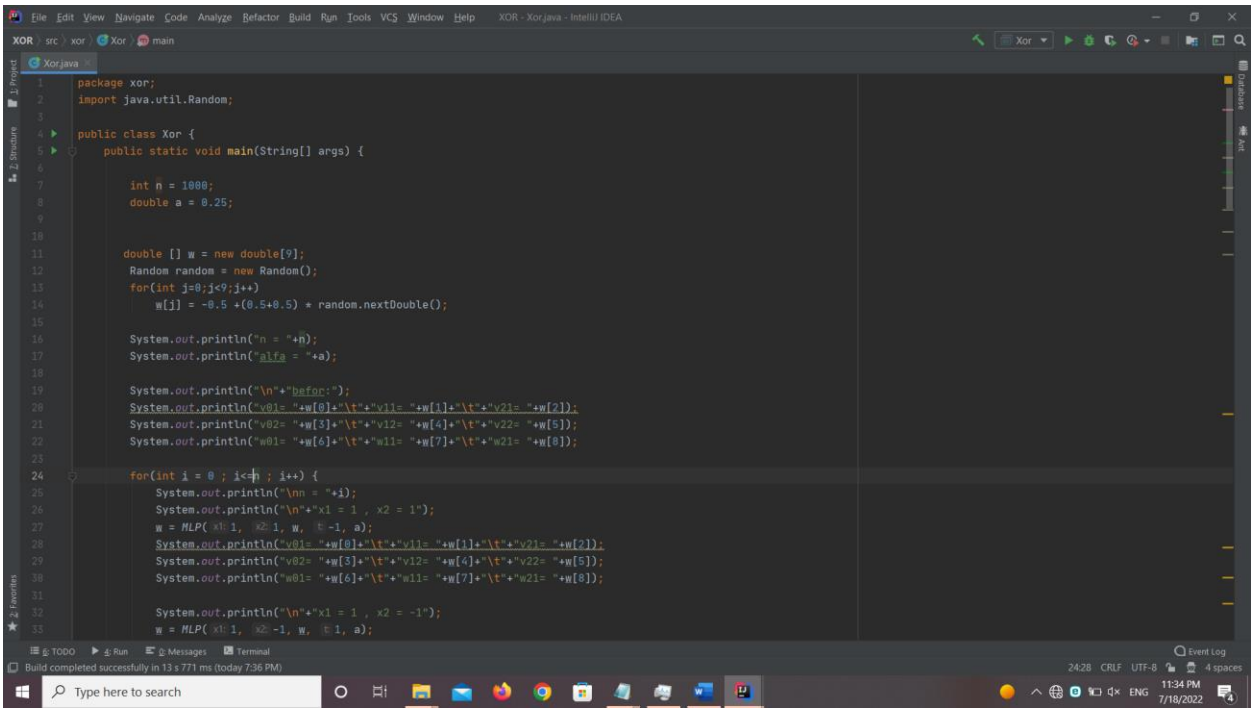
نرگس فیاض بخش 9815091026

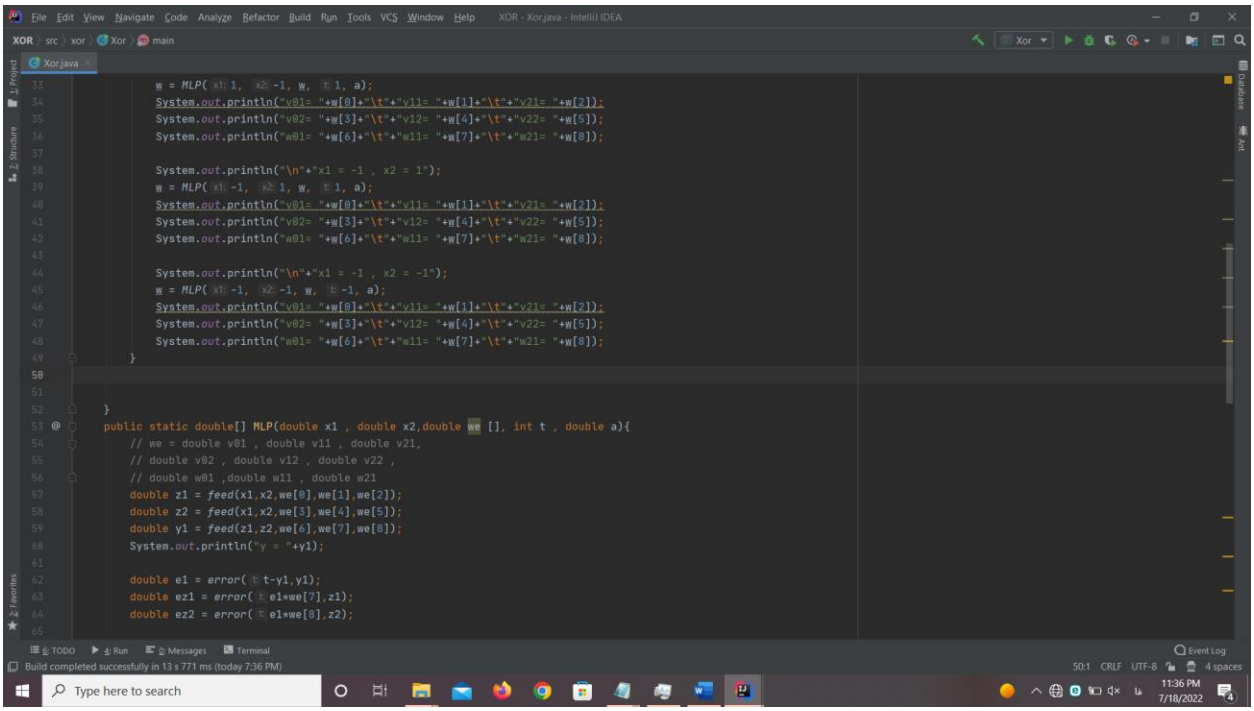
(1

Xor:

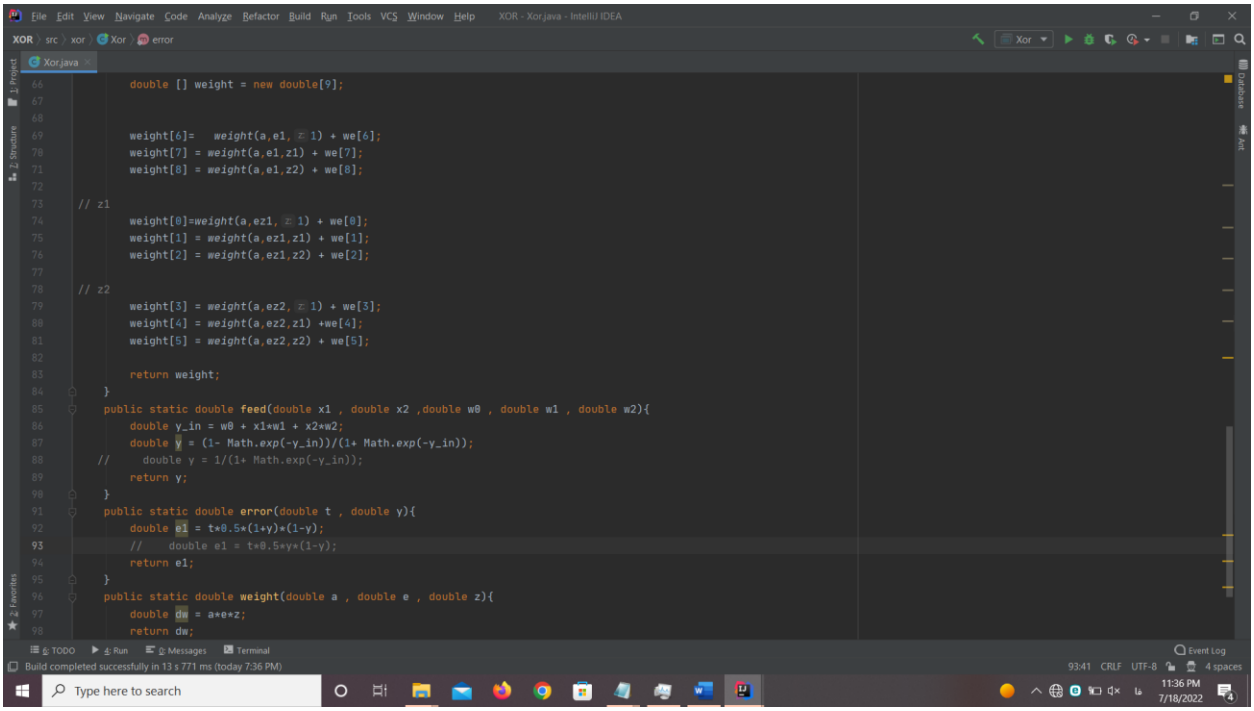
یک بخش از کد عملیات feed forward را انجام می دهد. بخش دیگر عملیات erro backward و بخش پایانی هم عملیات به روزرسانی وزن ها.

کد:

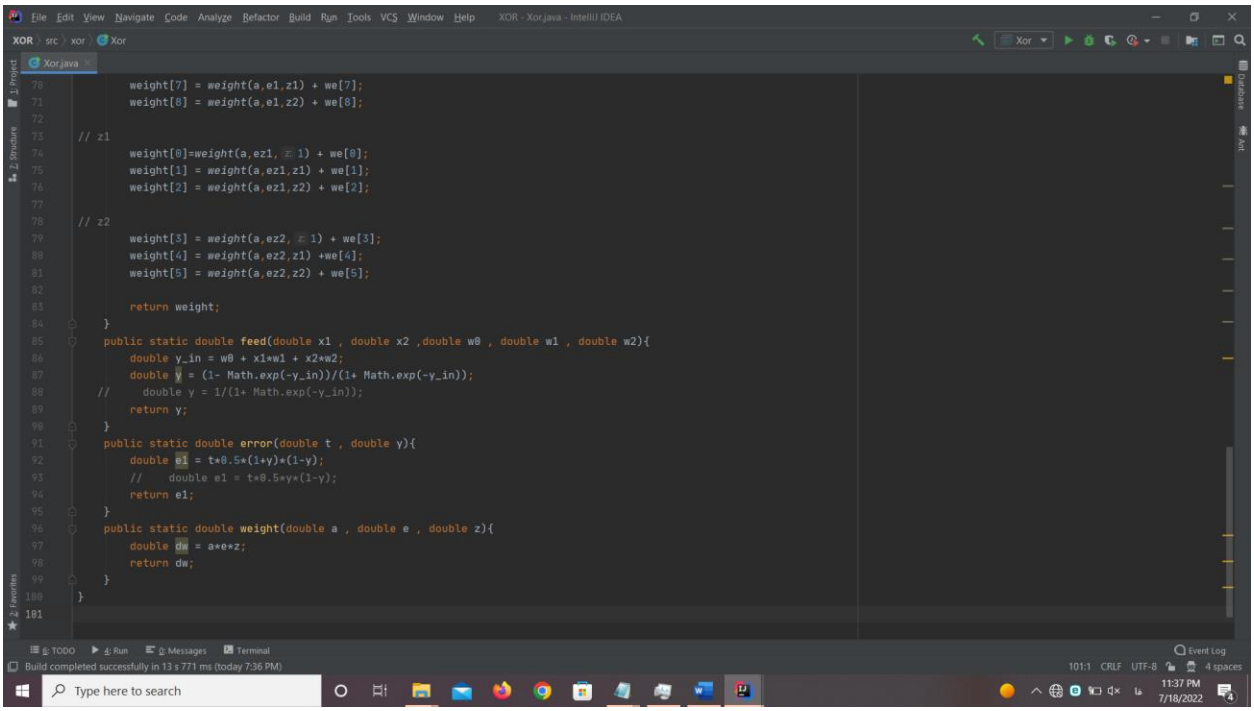




```
            w = MLP( x1: 1,  x2: -1, w,  t: 1, a);
            System.out.println("v01= "+w[0]+"\t"+"v11= "+w[1]+"\t"+"v21= "+w[2]);
            System.out.println("v02= "+w[3]+"\t"+"v12= "+w[4]+"\t"+"v22= "+w[5]);
            System.out.println("w01= "+w[6]+"\t"+"w11= "+w[7]+"\t"+"w21= "+w[8]);

            System.out.println("\n"+"x1 = -1 , x2 = 1");
            w = MLP( x1: -1,  x2: 1, w,  t: 1, a);
            System.out.println("v01= "+w[0]+"\t"+"v11= "+w[1]+"\t"+"v21= "+w[2]);
            System.out.println("v02= "+w[3]+"\t"+"v12= "+w[4]+"\t"+"v22= "+w[5]);
            System.out.println("w01= "+w[6]+"\t"+"w11= "+w[7]+"\t"+"w21= "+w[8]);

            System.out.println("\n"+"x1 = -1 , x2 = -1");
            w = MLP( x1: -1,  x2: -1, w,  t: -1, a);
            System.out.println("v01= "+w[0]+"\t"+"v11= "+w[1]+"\t"+"v21= "+w[2]);
            System.out.println("v02= "+w[3]+"\t"+"v12= "+w[4]+"\t"+"v22= "+w[5]);
            System.out.println("w01= "+w[6]+"\t"+"w11= "+w[7]+"\t"+"w21= "+w[8]);
        }


    }
    public static double[] MLP(double x1 , double x2,double we [], int t , double a){
        // we = double v01 , double v11 , double v21,
        // double v02 , double v12 , double v22 ,
        // double w01 ,double w11 , double w21
        double z1 = feed(x1,x2,we[0],we[1],we[2]);
        double z2 = feed(x1,x2,we[3],we[4],we[5]);
        double y1 = feed(z1,z2,we[6],we[7],we[8]);
        System.out.println("y = "+y1);

        double e1 = error( t: t-y1,y1);
        double ez1 = error( t: e1*we[7],z1);
        double ez2 = error( t: e1*we[8],z2);
```



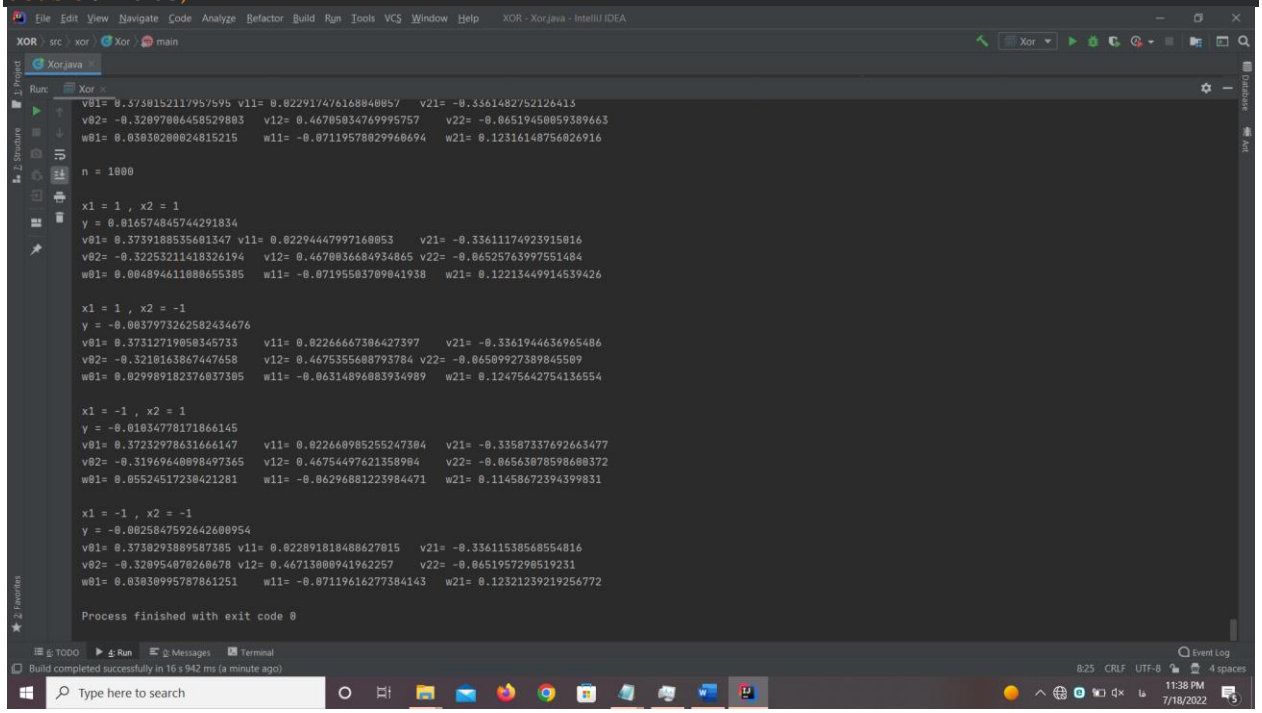
```
        double [] weight = new double[9];


        weight[6]=   weight(a,e1, z: 1) + we[6];
        weight[7] = weight(a,e1,z1) + we[7];
        weight[8] = weight(a,e1,z2) + we[8];

// z1
        weight[0]=weight(a,ez1, z: 1) + we[0];
        weight[1] = weight(a,ez1,z1) + we[1];
        weight[2] = weight(a,ez1,z2) + we[2];

// z2
        weight[3] = weight(a,ez2, z: 1) + we[3];
        weight[4] = weight(a,ez2,z1) +we[4];
        weight[5] = weight(a,ez2,z2) + we[5];

        return weight;
    }
    public static double feed(double x1 , double x2 ,double w0 , double w1 , double w2){
        double y_in = w0 + x1*w1 + x2*w2;
        double y = (1- Math.exp(-y_in))/(1+ Math.exp(-y_in));
    //     double y = 1/(1+ Math.exp(-y_in));
        return y;
    }
    public static double error(double t , double y){
        double e1 = t*0.5*(1+y)*(1-y);
        //     double e1 = t*0.5*y*(1-y);
        return e1;
    }
    public static double weight(double a , double e , double z){
        double dw = a*e*z;
        return dw;
```



```
        weight[7] = weight(a,e1,z1) + we[7];
        weight[8] = weight(a,e1,z2) + we[8];

// z1
        weight[0]=weight(a,ez1, z: 1) + we[0];
        weight[1] = weight(a,ez1,z1) + we[1];
        weight[2] = weight(a,ez1,z2) + we[2];

// z2
        weight[3] = weight(a,ez2, z: 1) + we[3];
        weight[4] = weight(a,ez2,z1) +we[4];
        weight[5] = weight(a,ez2,z2) + we[5];

        return weight;
    }
    public static double feed(double x1 , double x2 ,double w0 , double w1 , double w2){
        double y_in = w0 + x1*w1 + x2*w2;
        double y = (1- Math.exp(-y_in))/(1+ Math.exp(-y_in));
    //    double y = 1/(1+ Math.exp(-y_in));
        return y;
    }
    public static double error(double t , double y){
        double e1 = t*0.5*(1+y)*(1-y);
        //    double e1 = t*0.5*y*(1-y);
        return e1;
    }
    public static double weight(double a , double e , double z){
        double dw = a*e*z;
        return dw;
    }
}
```

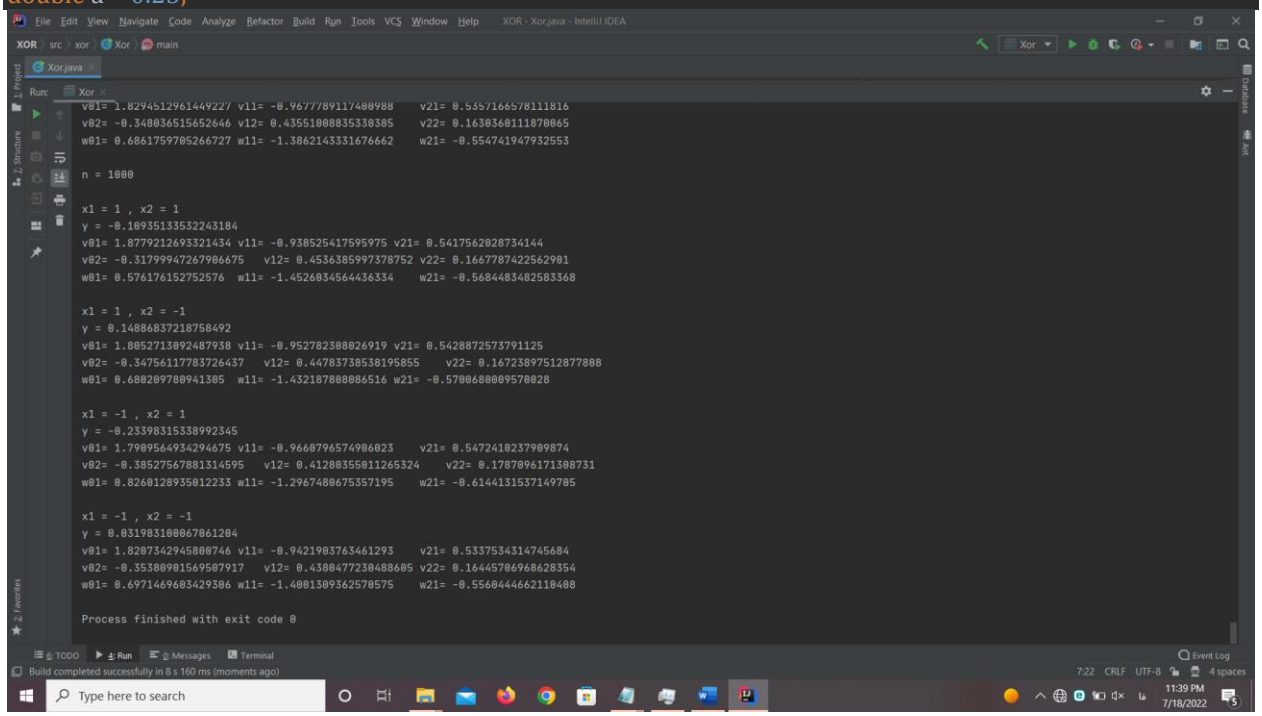
خروجی:

```
int n = 1000;
double a = 0.05;
```

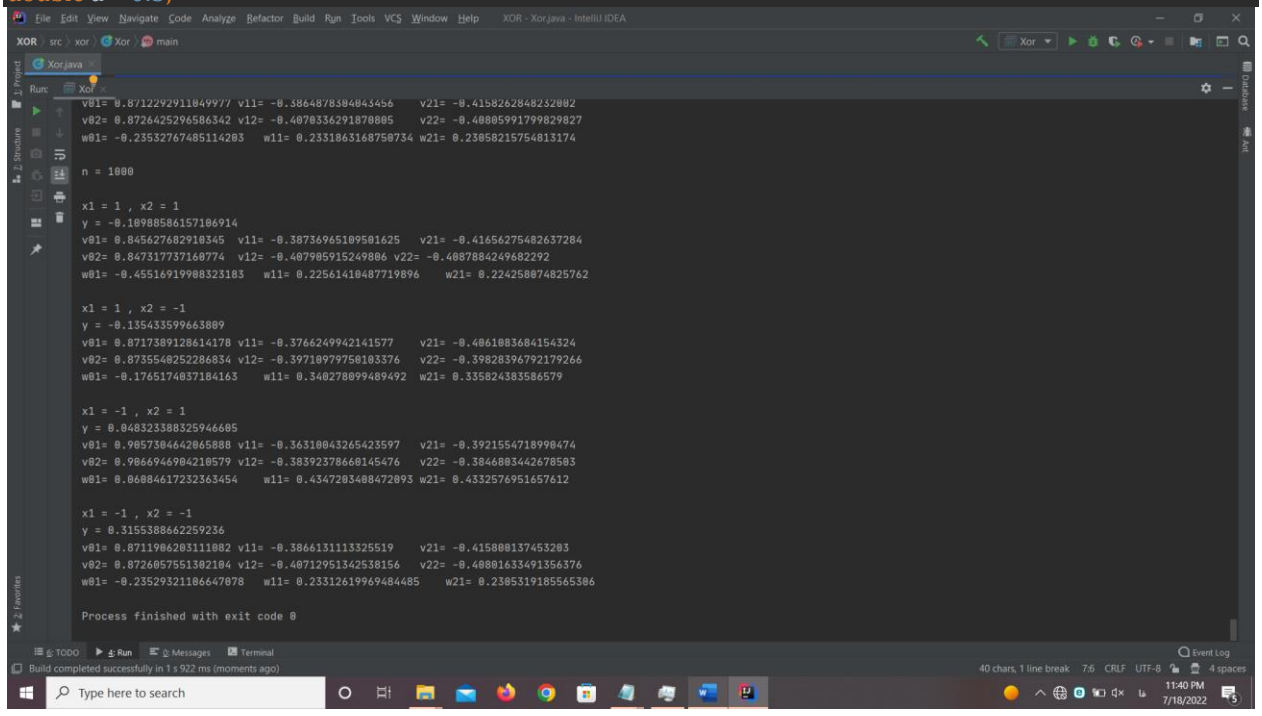


```
v01= 0.3738152117957595 v11= 0.022917476168040857   v21= -0.3361482752126413
v02= -0.32097006458529803   v12= 0.46705034769995757    v22= -0.06519450059389663
w01= 0.03030200024815215    w11= -0.07119578029960694   w21= 0.12316148756026916

n = 1000

x1 = 1 , x2 = 1
y = 0.016574845744291834
v01= 0.3739188535601347 v11= 0.02294447997160053    v21= -0.33611174923915016
v02= -0.32253211418326194   v12= 0.4670036684934865 v22= -0.06525763997551484
w01= 0.004894611080655385   w11= -0.07195503709041938   w21= 0.12213449914539426

x1 = 1 , x2 = -1
y = -0.0837973262582434676
v01= 0.37312719050345733    v11= 0.02266667306427397    v21= -0.3361944636965486
v02= -0.3210163867447658    v12= 0.4675355608793784 v22= -0.06509273898455509
w01= 0.029989182376037385   w11= -0.06314896083934989   w21= 0.12475642754136554

x1 = -1 , x2 = 1
y = -0.01034778171866145
v01= 0.37232978631666147    v11= 0.0226609852552473044   v21= -0.33587337692663477
v02= -0.31969640898497365   v12= 0.46754497621358904    v22= -0.06563078598600372
w01= 0.05524517230421281    w11= -0.06296881223984471   w21= 0.11458672394399831

x1 = -1 , x2 = -1
y = -0.0025847592642600954
v01= 0.3738293889587385 v11= 0.022891818488627015   v21= -0.33611538568554816
v02= -0.320954070260678 v12= 0.46713008941962257    v22= -0.0651957298519231
w01= 0.03030995787861251    w11= -0.07119616277384143   w21= 0.12321239219256772

Process finished with exit code 0
```

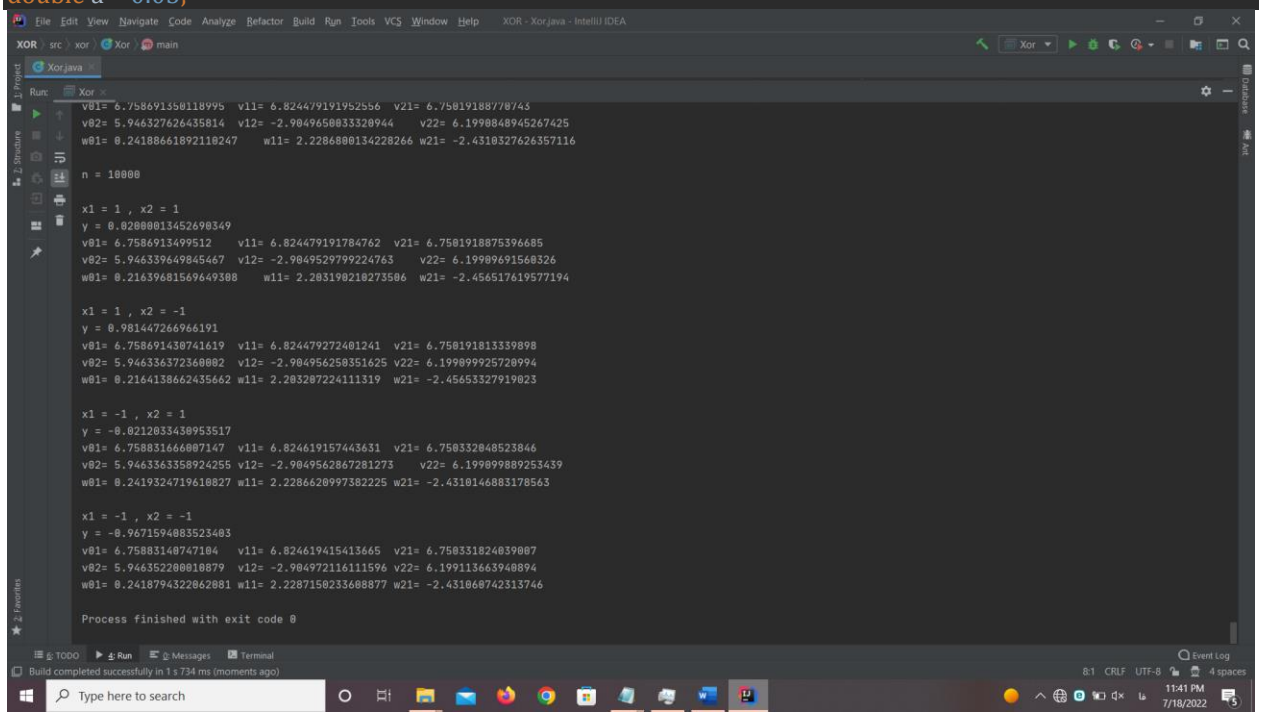
```
int n = 1000;
double a = 0.25;
```



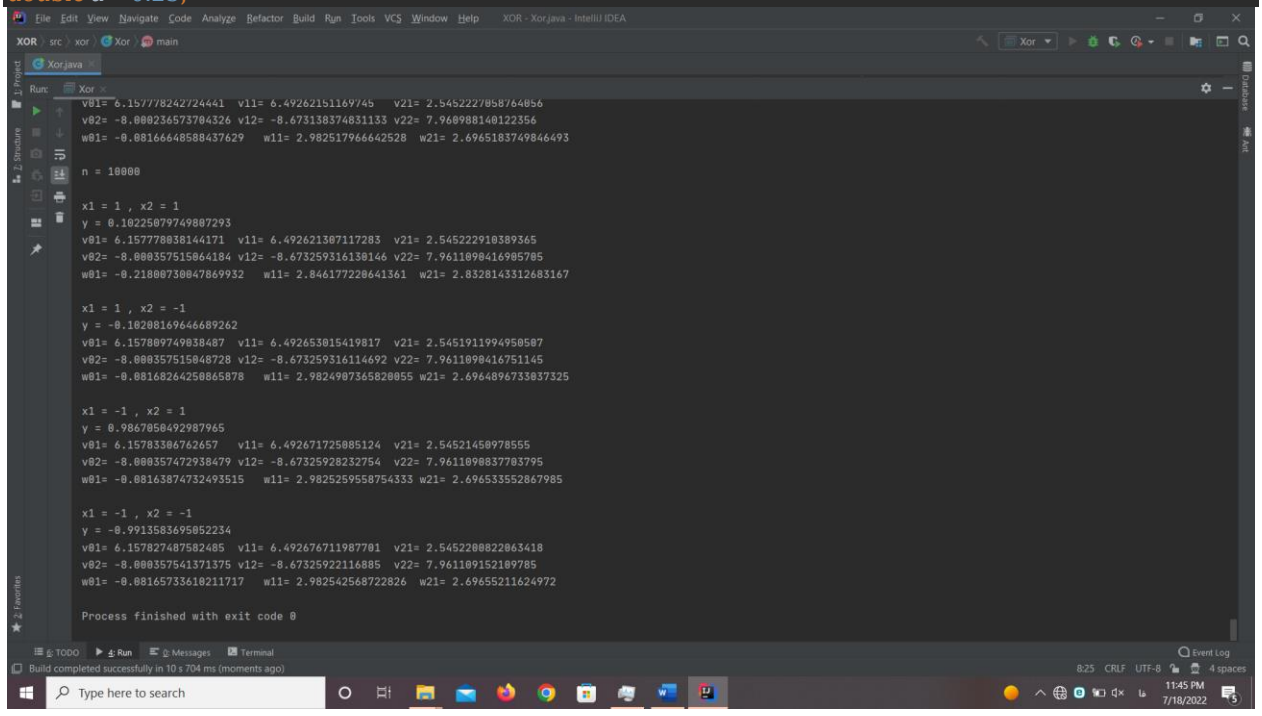

```
int n = 1000;
double a = 0.5;
```



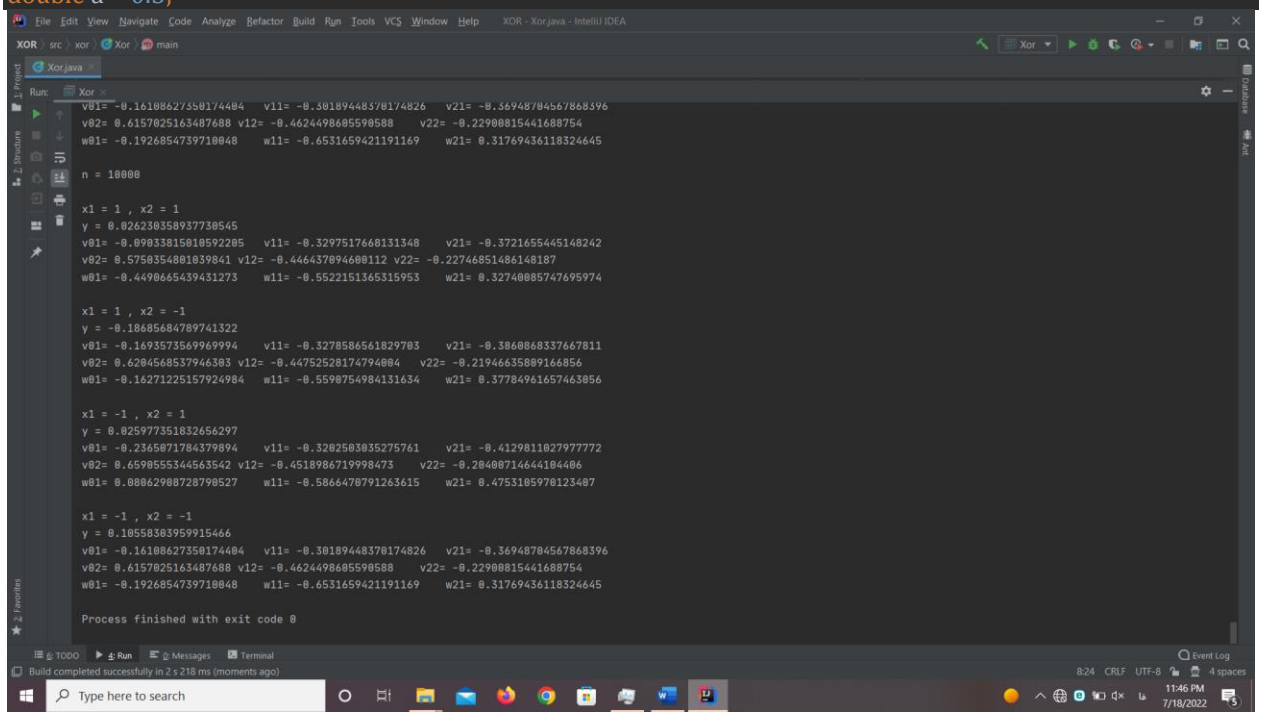
```
int n = 10000;
double a = 0.05;
```



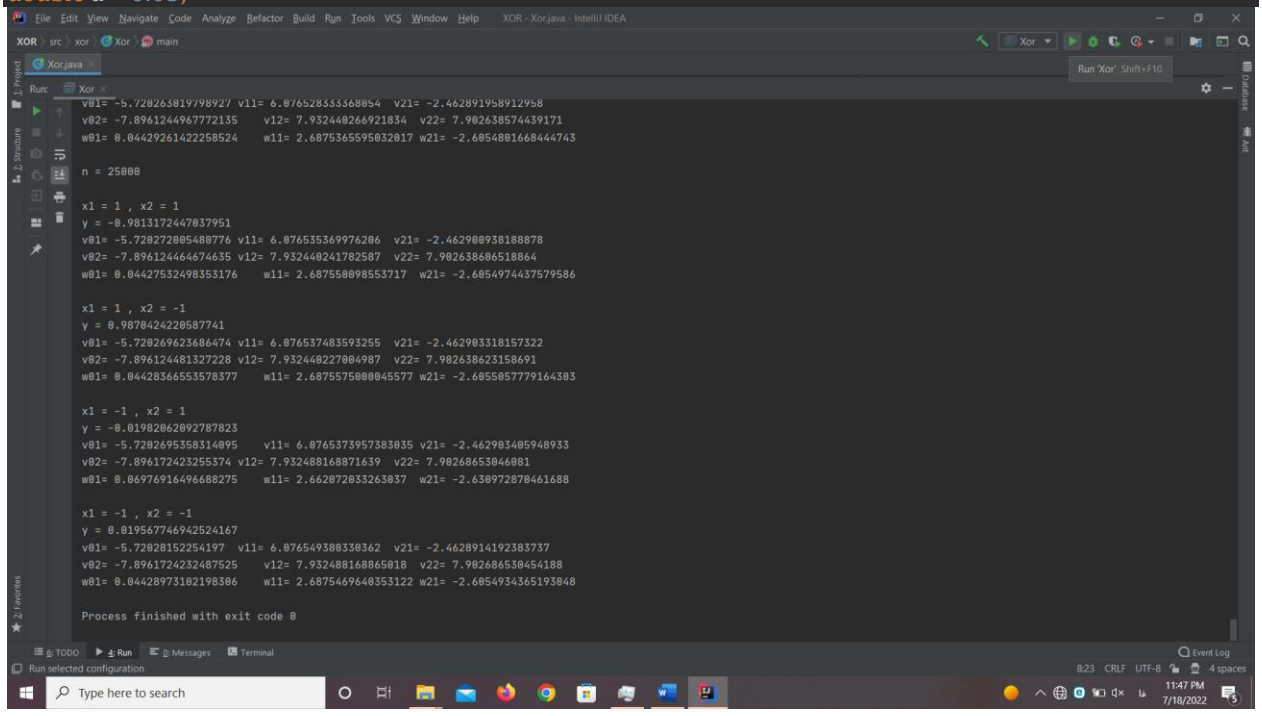

```
int n = 10000;
double a = 0.25;
```



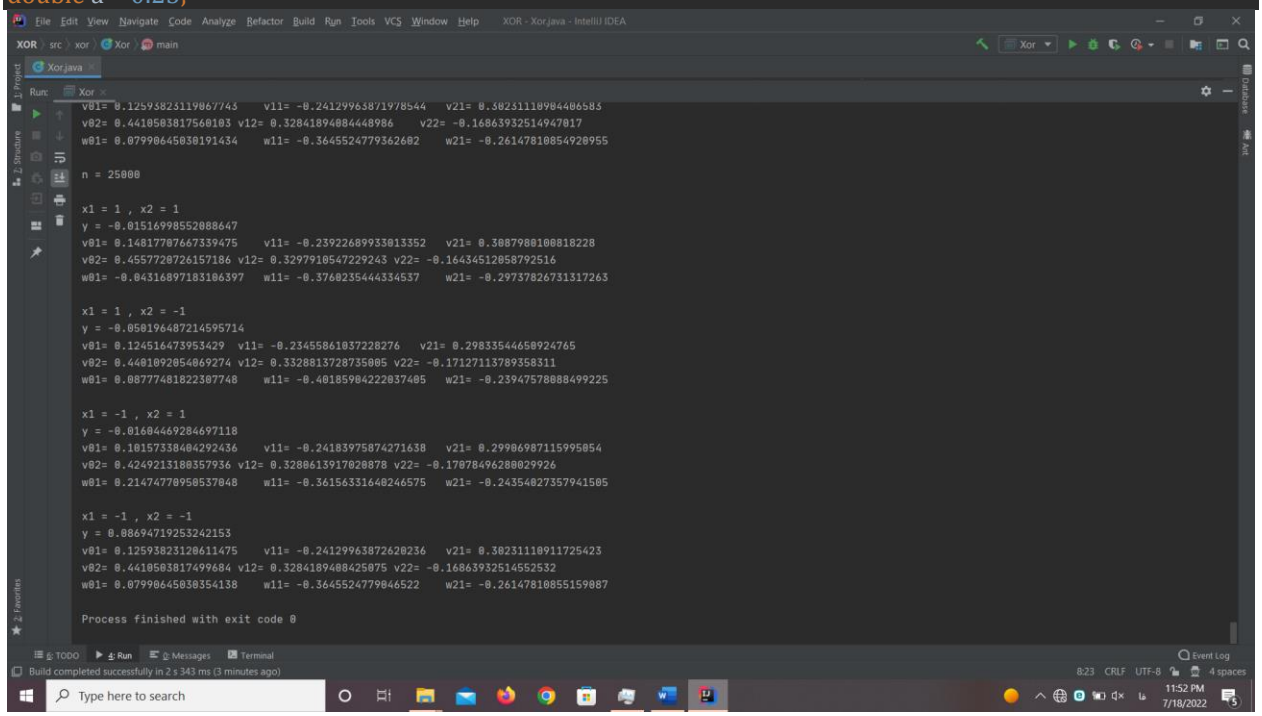
```
int n = 10000;
double a = 0.5;
```



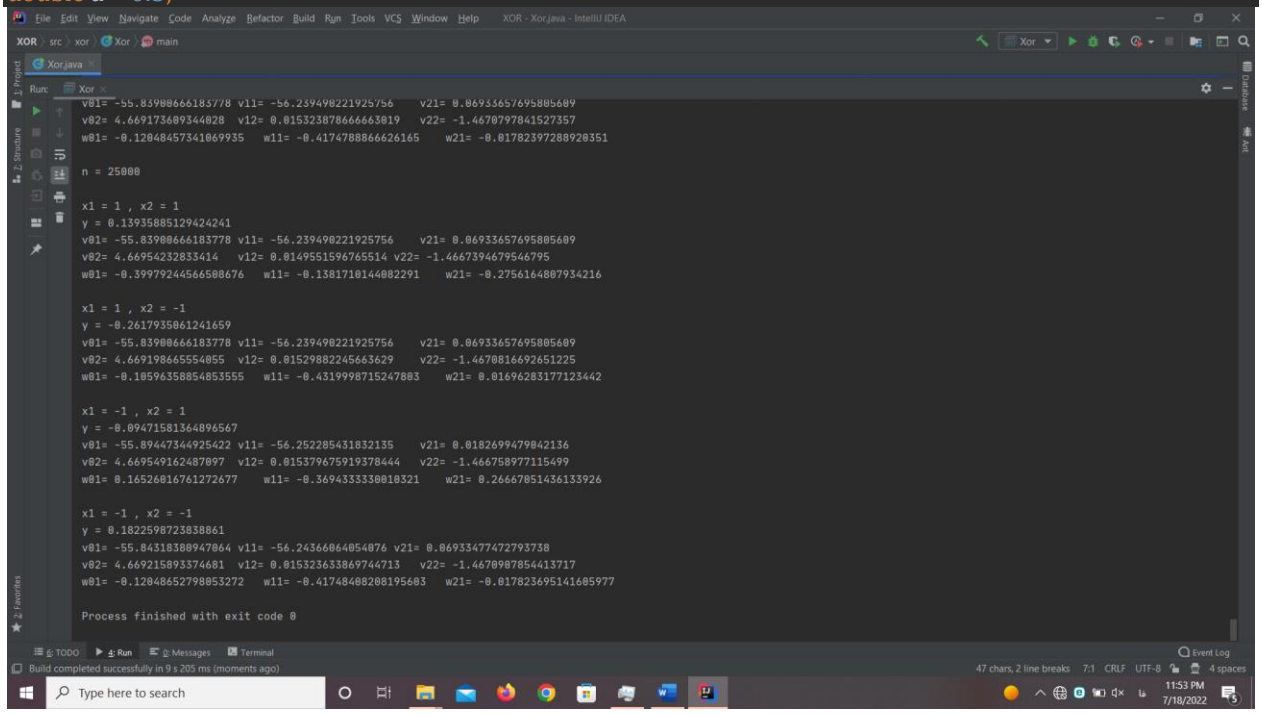

```
int n = 25000;
double a = 0.05;
```



```
int n = 25000;
double a = 0.25;
```



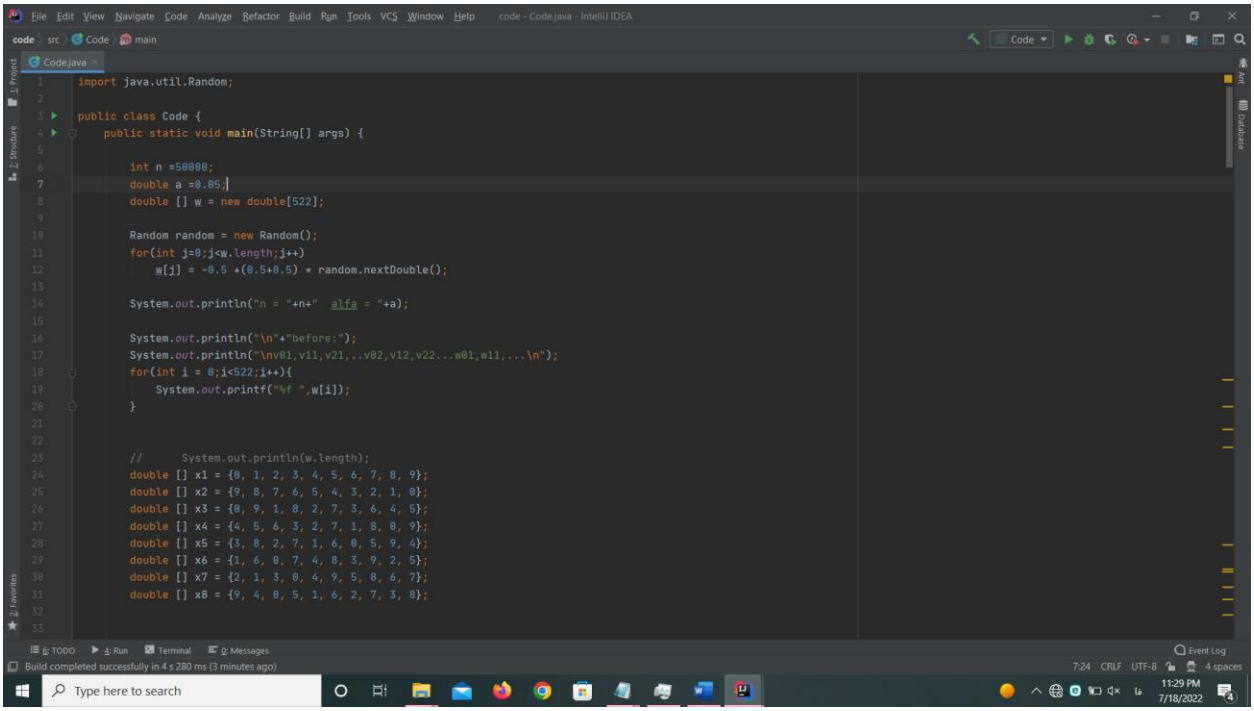

```
int n = 25000;
double a = 0.5;
```



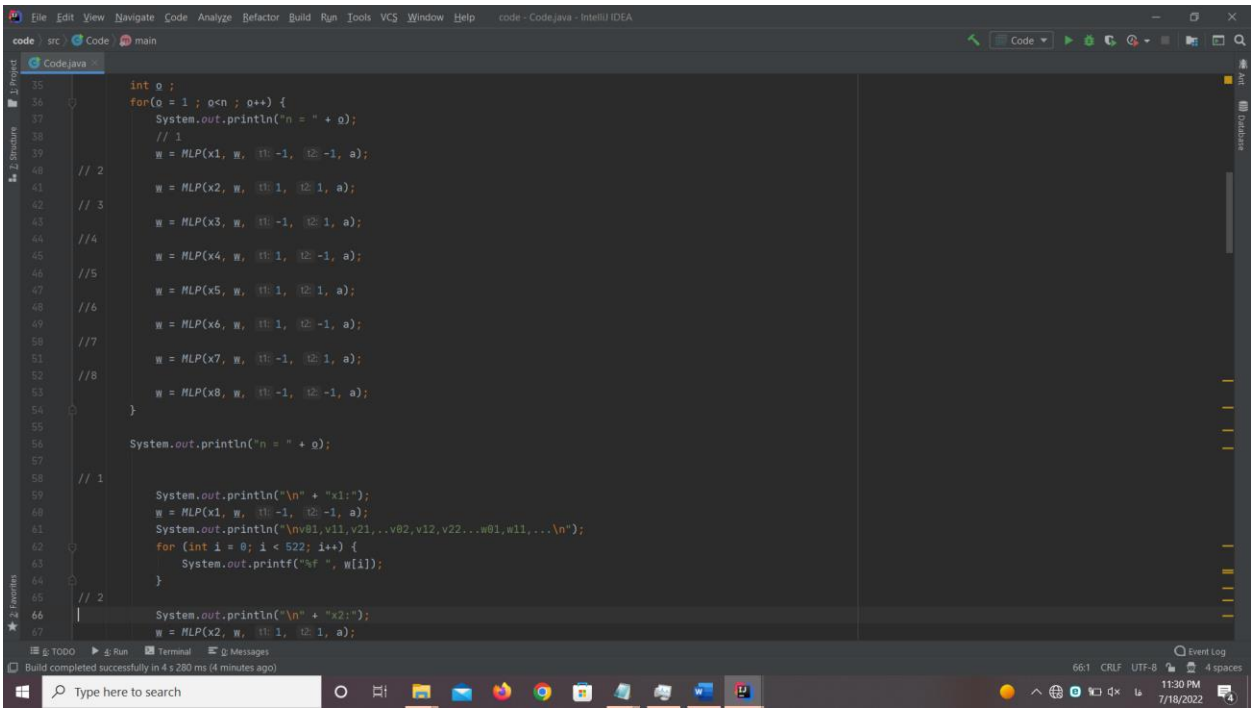
(2

این کد تعمیم یافته کد قبلی است و مراحل هم شبیه قبلی

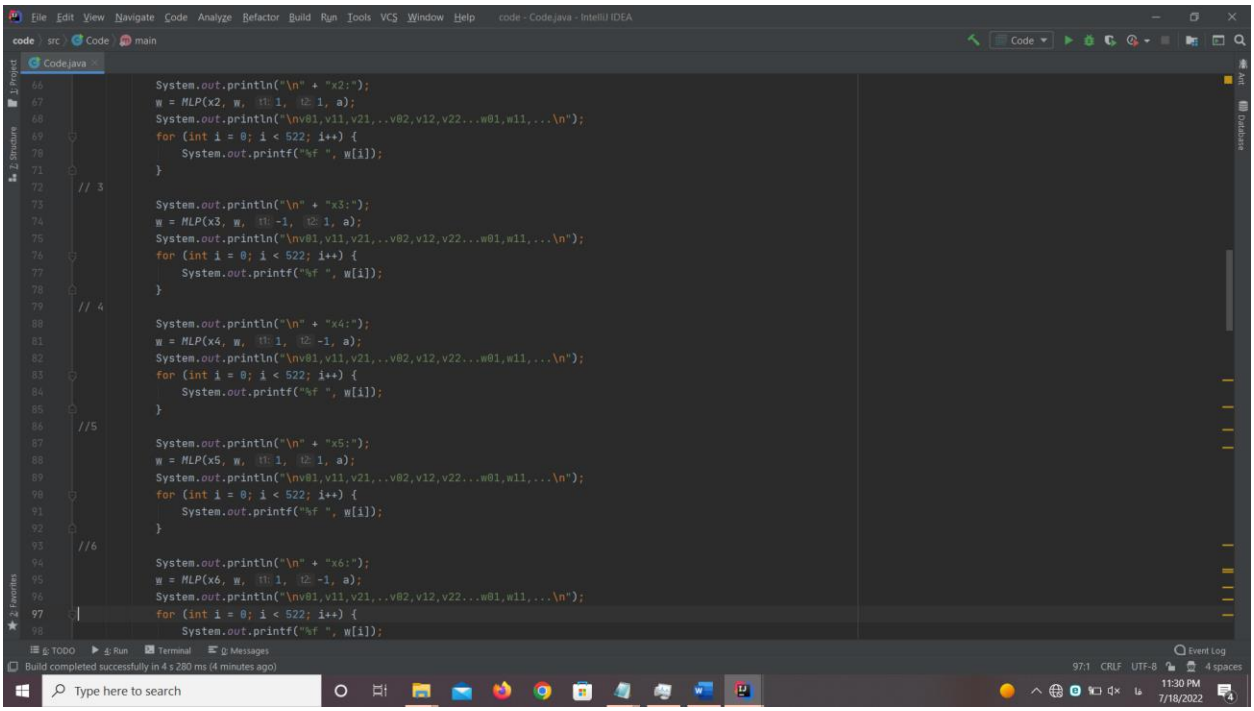
کد:



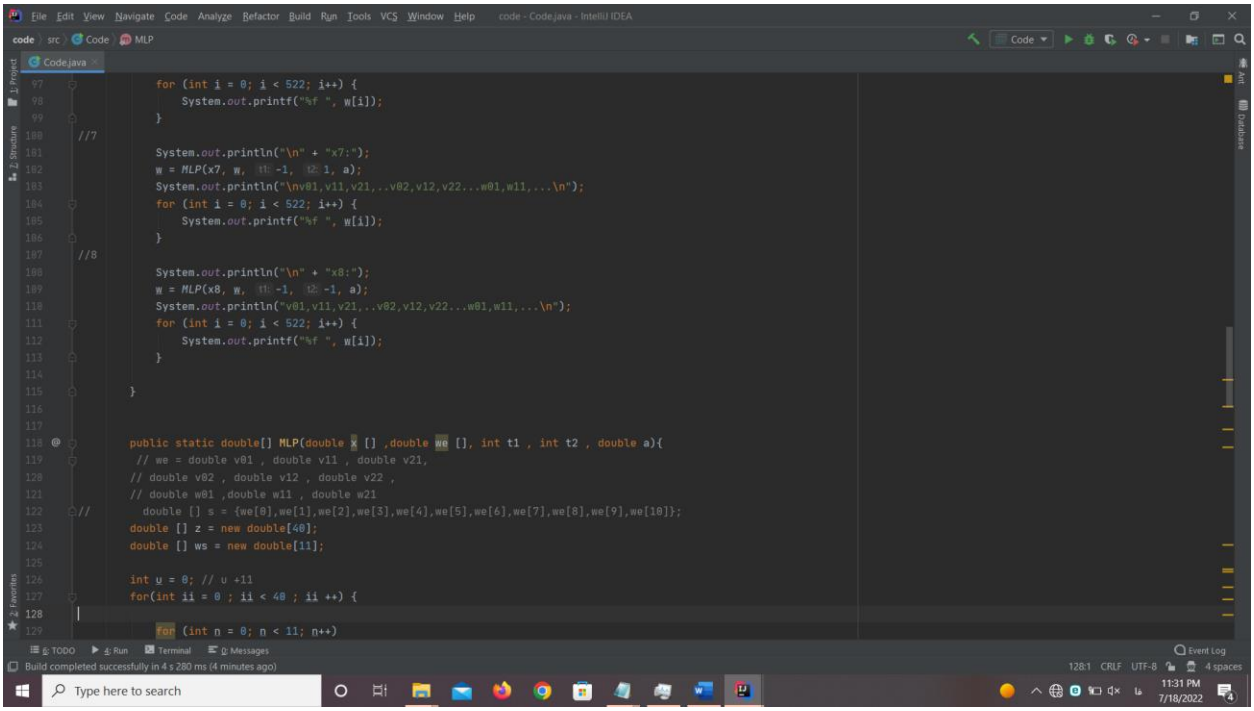
```java
import java.util.Random;

public class Code {
    public static void main(String[] args) {

        int n =50000;
        double a =0.05;
        double [] w = new double[522];

        Random random = new Random();
        for(int j=0;j<w.length;j++)
            w[j] = -0.5 +(0.5+0.5) * random.nextDouble();

        System.out.println("n = "+n+"  alfa = "+a);

        System.out.println("\n"+"before:");
        System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
        for(int i = 0;i<522;i++){
            System.out.printf("%f ",w[i]);
        }


        //      System.out.println(w.length);
        double [] x1 = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
        double [] x2 = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
        double [] x3 = {0, 9, 1, 8, 2, 7, 3, 6, 4, 5};
        double [] x4 = {4, 5, 6, 3, 2, 7, 1, 8, 0, 9};
        double [] x5 = {3, 8, 2, 7, 1, 6, 0, 5, 9, 4};
        double [] x6 = {1, 6, 0, 7, 4, 8, 3, 9, 2, 5};
        double [] x7 = {2, 1, 3, 0, 4, 9, 5, 8, 6, 7};
        double [] x8 = {9, 4, 0, 5, 1, 6, 2, 7, 3, 8};
```



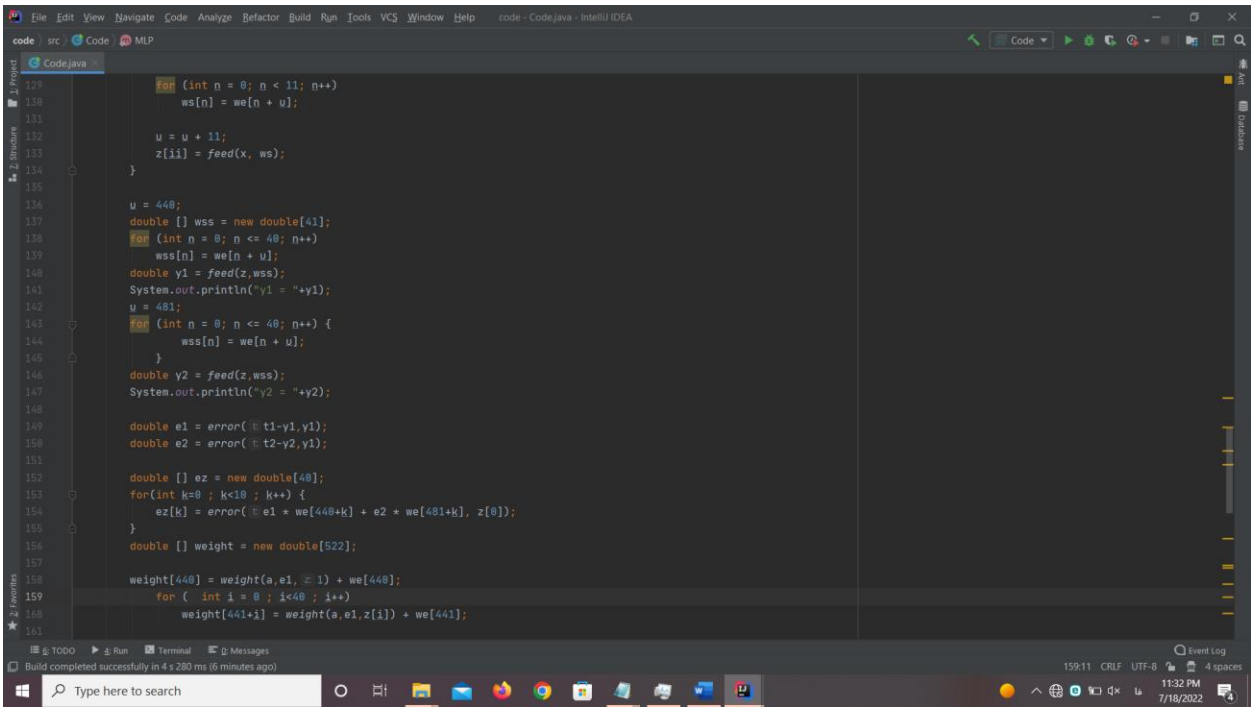
```java
        int o ;
        for(o = 1 ; o<n ; o++) {
            System.out.println("n = " + o);
            // 1
            w = MLP(x1, w,  t1: -1,  t2: -1, a);
// 2
            w = MLP(x2, w,  t1: 1,  t2: 1, a);
// 3
            w = MLP(x3, w,  t1: -1,  t2: 1, a);
//4
            w = MLP(x4, w,  t1: 1,  t2: -1, a);
//5
            w = MLP(x5, w,  t1: 1,  t2: 1, a);
//6
            w = MLP(x6, w,  t1: 1,  t2: -1, a);
//7
            w = MLP(x7, w,  t1: -1,  t2: 1, a);
//8
            w = MLP(x8, w,  t1: -1,  t2: -1, a);
        }

        System.out.println("n = " + o);


// 1

            System.out.println("\n" + "x1:");
            w = MLP(x1, w,  t1: -1,  t2: -1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
// 2
            System.out.println("\n" + "x2:");
            w = MLP(x2, w,  t1: 1,  t2: 1, a);
```



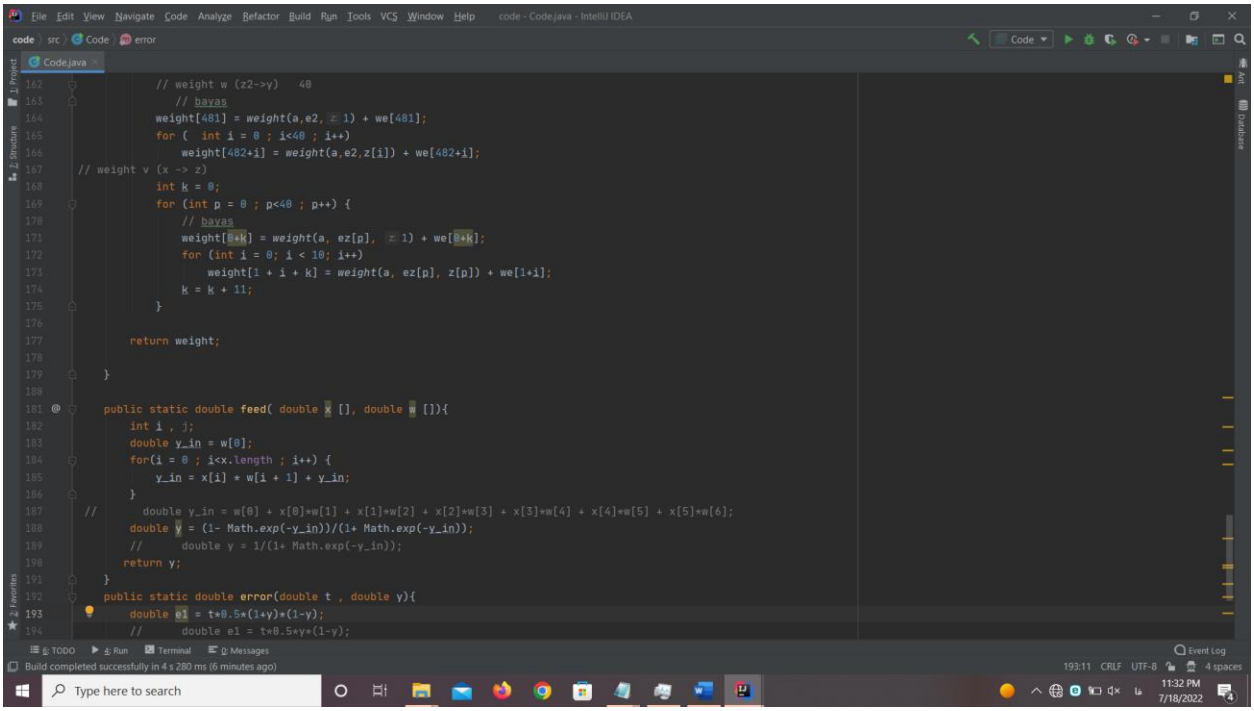
```
            System.out.println("\n" + "x2:");
            w = MLP(x2, w, t1: 1, t2: 1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
// 3
            System.out.println("\n" + "x3:");
            w = MLP(x3, w, t1: -1, t2: 1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
// 4
            System.out.println("\n" + "x4:");
            w = MLP(x4, w, t1: 1, t2: -1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
//5
            System.out.println("\n" + "x5:");
            w = MLP(x5, w, t1: 1, t2: 1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
//6
            System.out.println("\n" + "x6:");
            w = MLP(x6, w, t1: 1, t2: -1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
```



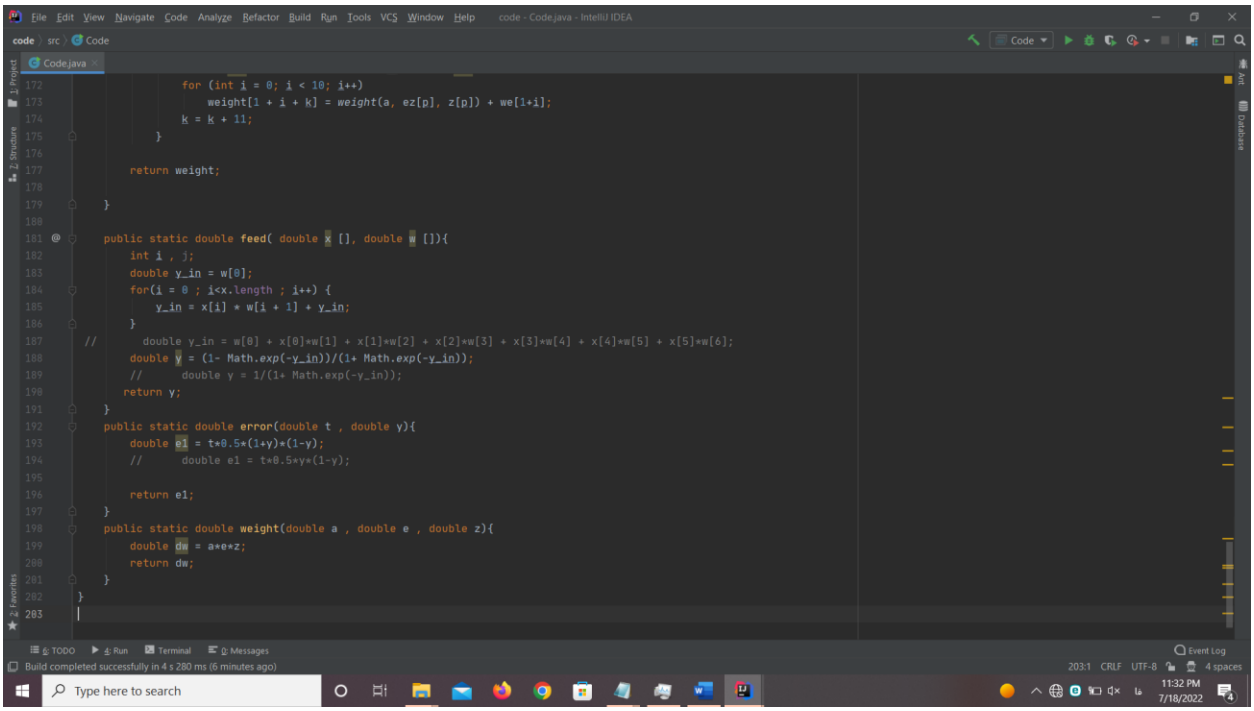
```
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
//7
            System.out.println("\n" + "x7:");
            w = MLP(x7, w, t1: -1, t2: 1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }
//8
            System.out.println("\n" + "x8:");
            w = MLP(x8, w, t1: -1, t2: -1, a);
            System.out.println("\nv01,v11,v21,..v02,v12,v22...w01,w11,...\n");
            for (int i = 0; i < 522; i++) {
                System.out.printf("%f ", w[i]);
            }

        }


        public static double[] MLP(double x [] ,double we [], int t1 , int t2 , double a){
         // we = double v01 , double v11 , double v21,
        // double v02 , double v12 , double v22 ,
        // double w01 ,double w11 , double w21
//          double [] s = {we[0],we[1],we[2],we[3],we[4],we[5],we[6],we[7],we[8],we[9],we[10]};
        double [] z = new double[40];
        double [] ws = new double[11];

        int u = 0; // u +11
        for(int ii = 0 ; ii < 40 ; ii ++) {

            for (int n = 0; n < 11; n++)
```



```
            for (int n = 0; n < 11; n++)
                ws[n] = we[n + u];

            u = u + 11;
            z[ii] = feed(x, ws);
        }

        u = 440;
        double [] wss = new double[41];
        for (int n = 0; n <= 40; n++)
            wss[n] = we[n + u];
        double y1 = feed(z,wss);
        System.out.println("y1 = "+y1);
        u = 481;
        for (int n = 0; n <= 40; n++) {
                wss[n] = we[n + u];
            }
        double y2 = feed(z,wss);
        System.out.println("y2 = "+y2);

        double e1 = error( t: t1-y1,y1);
        double e2 = error( t: t2-y2,y1);

        double [] ez = new double[40];
        for(int k=0 ; k<10 ; k++) {
            ez[k] = error( t: e1 * we[440+k] + e2 * we[481+k], z[0]);
        }
        double [] weight = new double[522];

        weight[440] = weight(a,e1, z: 1) + we[440];
            for (  int i = 0 ; i<40 ; i++)
                weight[441+i] = weight(a,e1,z[i]) + we[441];
```



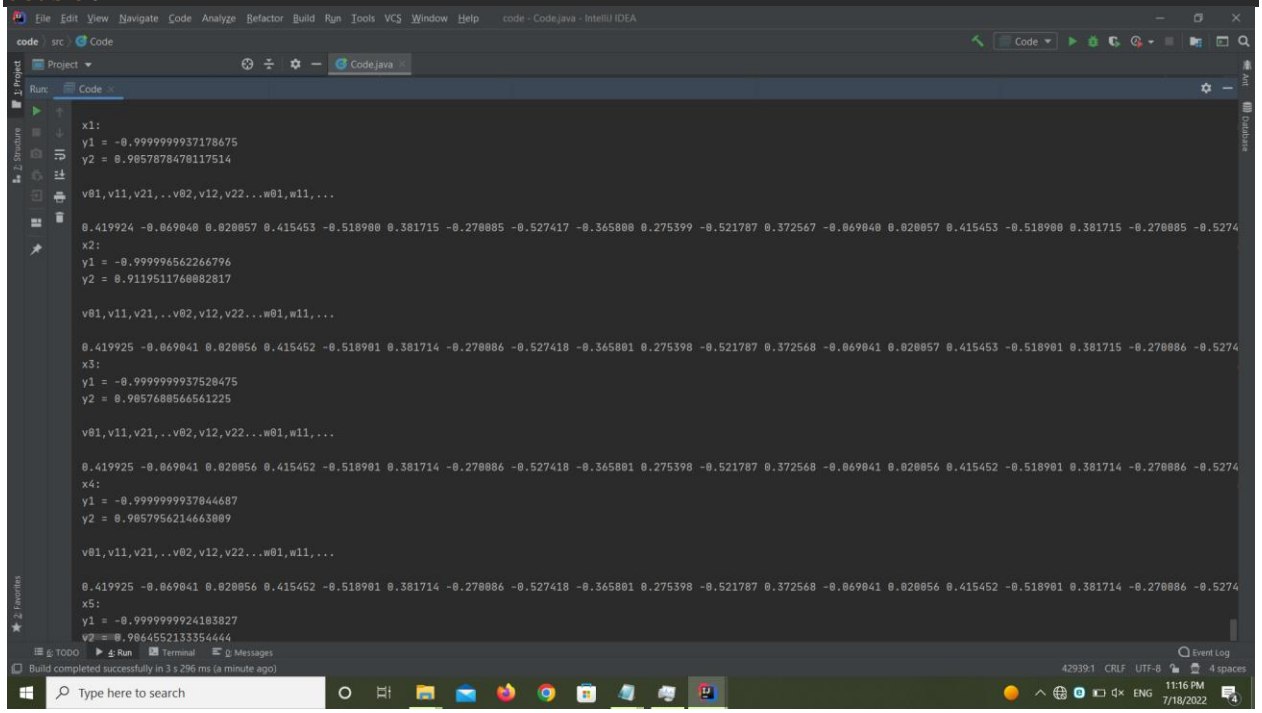
```
            // weight w (z2->y)   40
               // bayas
            weight[481] = weight(a,e2, z: 1) + we[481];
            for (  int i = 0 ; i<40 ; i++)
                weight[482+i] = weight(a,e2,z[i]) + we[482+i];
// weight v (x -> z)
            int k = 0;
            for (int p = 0 ; p<40 ; p++) {
                // bayas
                weight[0+k] = weight(a, ez[p], z: 1) + we[0+k];
                for (int i = 0; i < 10; i++)
                    weight[1 + i + k] = weight(a, ez[p], z[p]) + we[1+i];
                k = k + 11;
            }

        return weight;

    }

    public static double feed( double x [], double w []){
        int i , j;
        double y_in = w[0];
        for(i = 0 ; i<x.length ; i++) {
            y_in = x[i] * w[i + 1] + y_in;
        }
//        double y_in = w[0] + x[0]*w[1] + x[1]*w[2] + x[2]*w[3] + x[3]*w[4] + x[4]*w[5] + x[5]*w[6];
        double y = (1- Math.exp(-y_in))/(1+ Math.exp(-y_in));
        //      double y = 1/(1+ Math.exp(-y_in));
       return y;
    }
    public static double error(double t , double y){
        double e1 = t*0.5*(1+y)*(1-y);
        //      double e1 = t*0.5*y*(1-y);
```



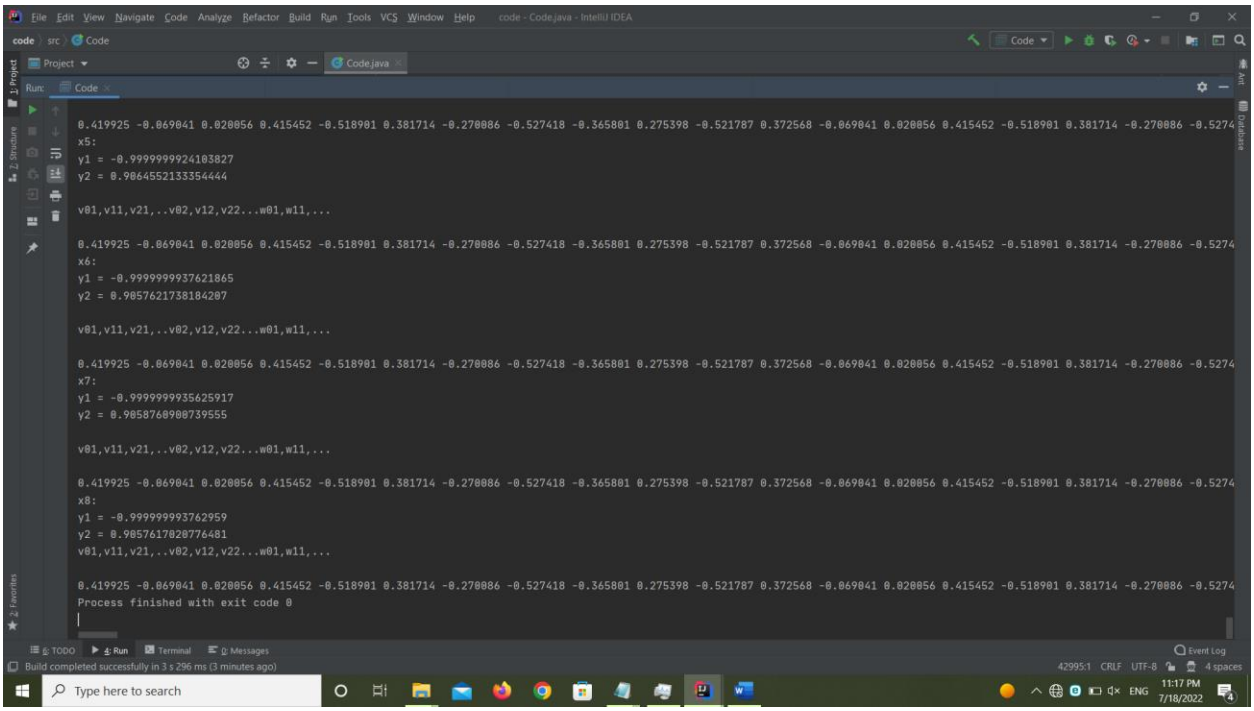
```
                for (int i = 0; i < 10; i++)
                    weight[1 + i + k] = weight(a, ez[p], z[p]) + we[1+i];
                k = k + 11;
            }

        return weight;

    }

    public static double feed( double x [], double w []){
        int i , j;
        double y_in = w[0];
        for(i = 0 ; i<x.length ; i++) {
            y_in = x[i] * w[i + 1] + y_in;
        }
//        double y_in = w[0] + x[0]*w[1] + x[1]*w[2] + x[2]*w[3] + x[3]*w[4] + x[4]*w[5] + x[5]*w[6];
        double y = (1- Math.exp(-y_in))/(1+ Math.exp(-y_in));
        //      double y = 1/(1+ Math.exp(-y_in));
      return y;
    }
    public static double error(double t , double y){
        double e1 = t*0.5*(1+y)*(1-y);
        //      double e1 = t*0.5*y*(1-y);

        return e1;
    }
    public static double weight(double a , double e , double z){
        double dw = a*e*z;
        return dw;
    }
}
```

خروجی:

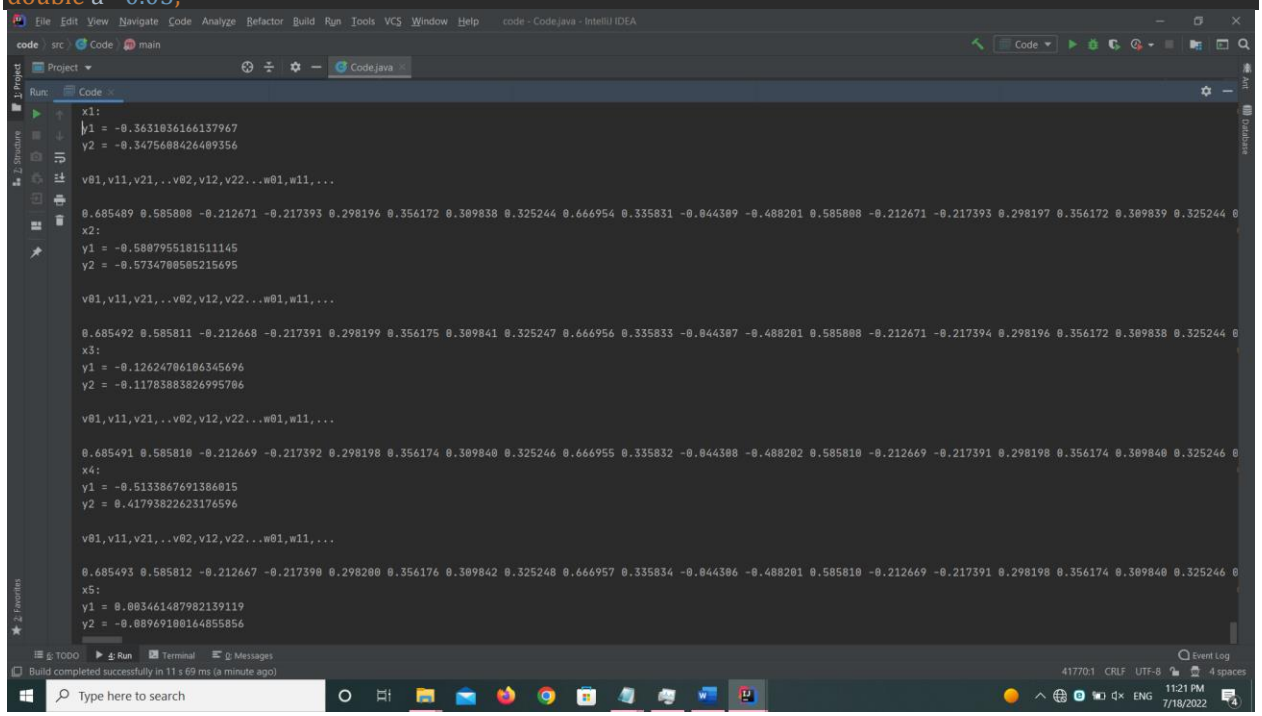
```
int n =5000;
double a
```

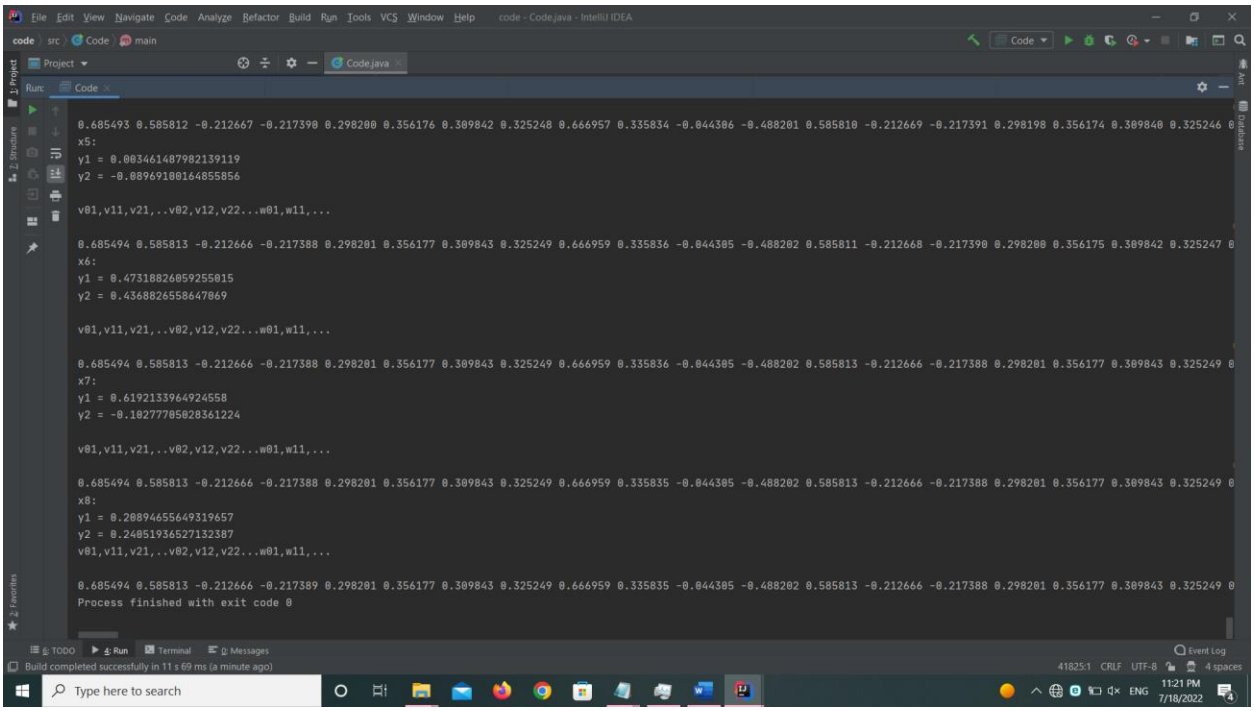


```
x1:
y1 = -0.9999999937178675
y2 = 0.9057878478117514

v01,v11,v21,..v02,v12,v22...w01,w11,...

0.419924 -0.869040 0.020057 0.415453 -0.518900 0.381715 -0.270885 -0.527417 -0.365800 0.275399 -0.521787 0.372567 -0.869040 0.020057 0.415453 -0.518900 0.381715 -0.270885 -0.5274
x2:
y1 = -0.999996562266796
y2 = 0.9119511760882817

v01,v11,v21,..v02,v12,v22...w01,w11,...

0.419925 -0.869041 0.020056 0.415452 -0.518901 0.381714 -0.270886 -0.527418 -0.365801 0.275398 -0.521787 0.372568 -0.869041 0.020057 0.415453 -0.518901 0.381715 -0.270886 -0.5274
x3:
y1 = -0.9999999937528475
y2 = 0.9057680566561225

v01,v11,v21,..v02,v12,v22...w01,w11,...

0.419925 -0.869041 0.020056 0.415452 -0.518901 0.381714 -0.270886 -0.527418 -0.365801 0.275398 -0.521787 0.372568 -0.869041 0.020056 0.415452 -0.518901 0.381714 -0.270886 -0.5274
x4:
y1 = -0.9999999937844687
y2 = 0.9057956214663089

v01,v11,v21,..v02,v12,v22...w01,w11,...

0.419925 -0.869041 0.020056 0.415452 -0.518901 0.381714 -0.270886 -0.527418 -0.365801 0.275398 -0.521787 0.372568 -0.869041 0.020056 0.415452 -0.518901 0.381714 -0.270886 -0.5274
x5:
y1 = -0.9999999924103827
y2 = 0.9064552133354444
```

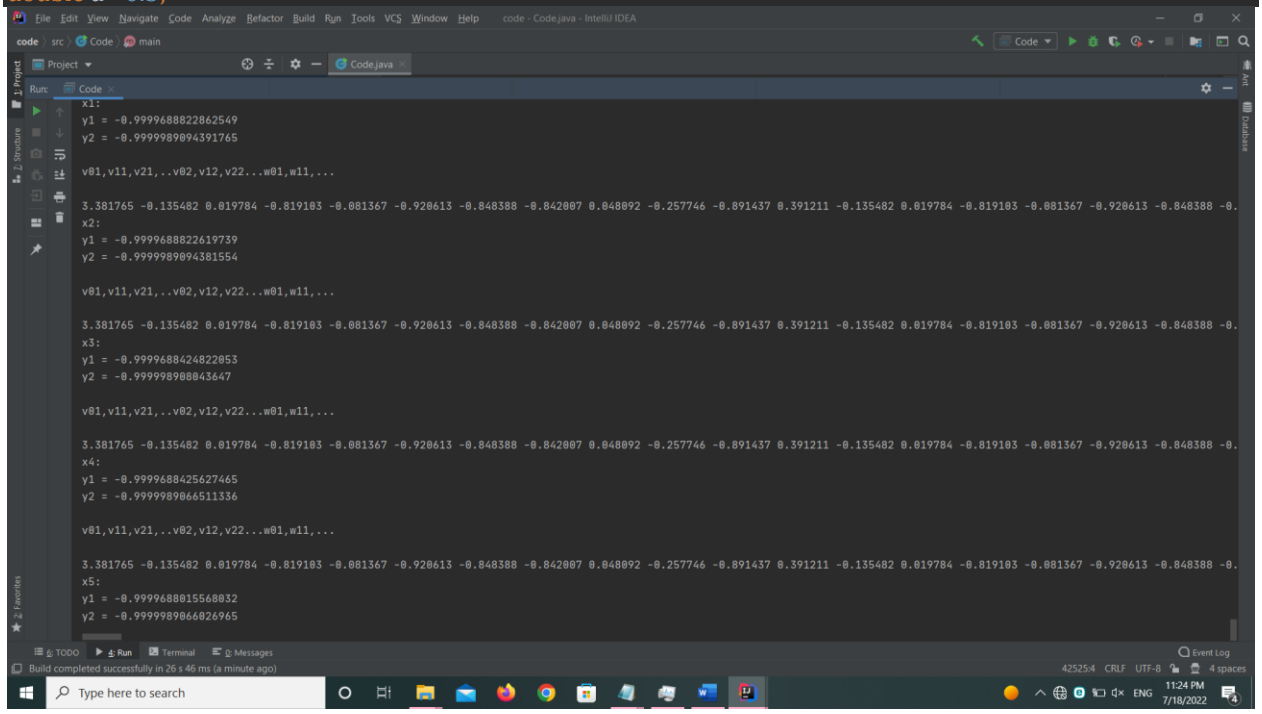


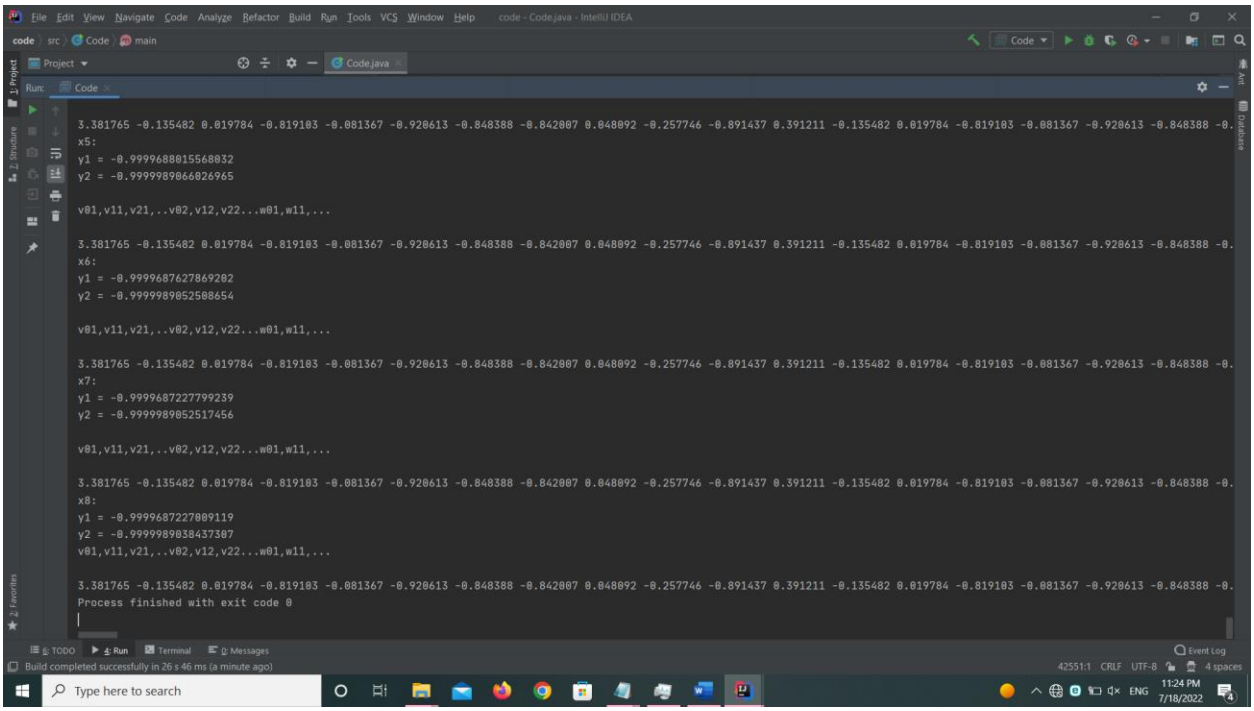

```
int n =5000;
double a =0.05;
```



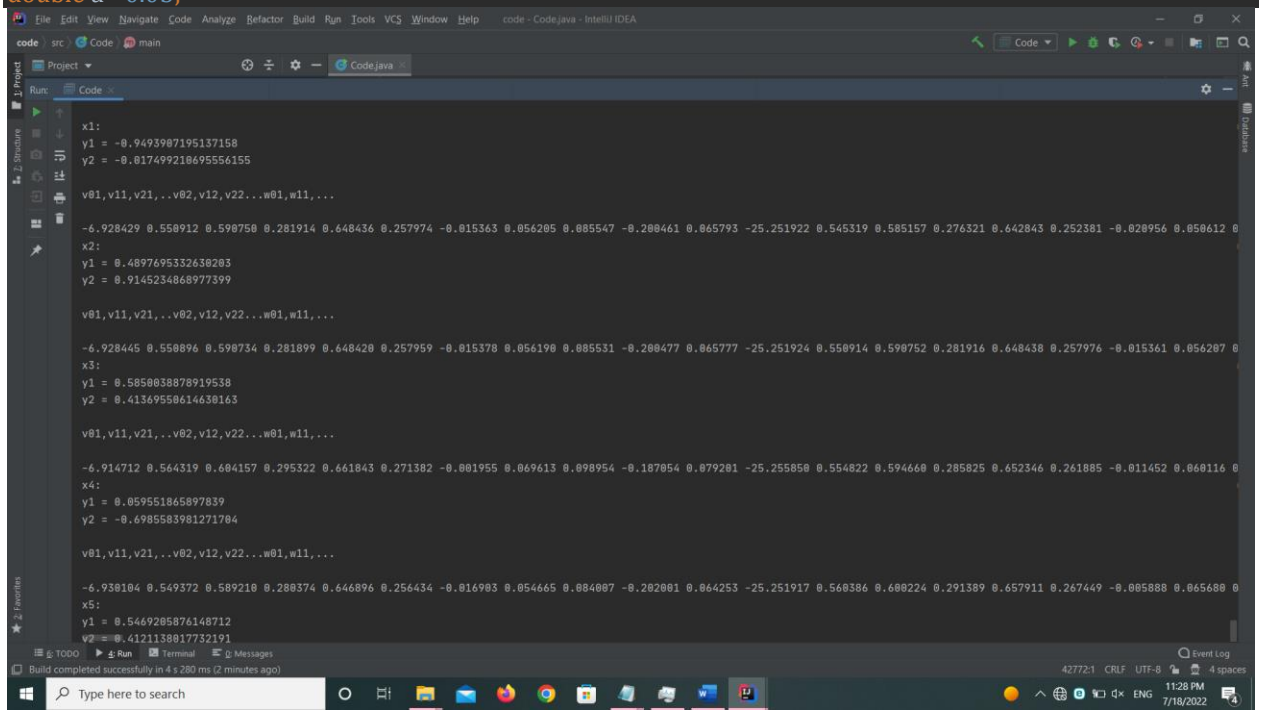


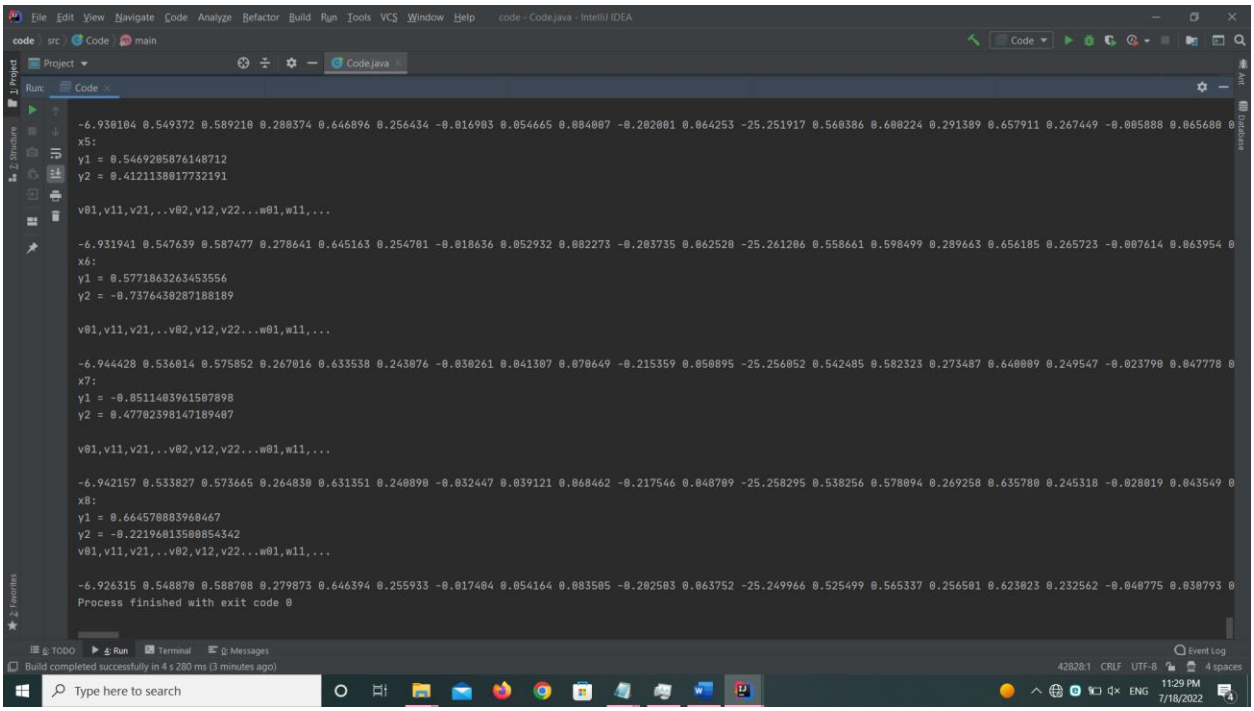

```
int n =50000;
double a =0.5;
```






```
int n =50000;
double a =0.05;
```




File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
code - Code.java - IntelliJ IDEA
code src Code main
Project
Code.java
Run: Code
-6.930104 0.549372 0.589210 0.280374 0.646896 0.256434 -0.016903 0.054665 0.084007 -0.202001 0.864253 -25.251917 0.560386 0.680224 0.291389 0.657911 0.267449 -0.005888 0.865680 0
x5:
y1 = 0.5469205876148712
y2 = 0.4121138017732191
v01,v11,v21,..v02,v12,v22...w01,w11,...
-6.931941 0.547639 0.587477 0.278641 0.645163 0.254701 -0.018636 0.052932 0.082273 -0.203735 0.862520 -25.261206 0.558661 0.598499 0.289663 0.656185 0.265723 -0.007614 0.863954 0
x6:
y1 = 0.5771863263453556
y2 = -0.7376430287188189
v01,v11,v21,..v02,v12,v22...w01,w11,...
-6.944428 0.536014 0.575852 0.267016 0.633538 0.243076 -0.030261 0.041307 0.070649 -0.215359 0.850895 -25.256052 0.542485 0.582323 0.273487 0.640089 0.249547 -0.023790 0.847778 0
x7:
y1 = -0.8511403961507898
y2 = 0.47782398147189407
v01,v11,v21,..v02,v12,v22...w01,w11,...
-6.942157 0.533827 0.573665 0.264830 0.631351 0.240898 -0.032447 0.039121 0.068462 -0.217546 0.848789 -25.258295 0.538256 0.578094 0.269258 0.635780 0.245318 -0.028019 0.843549 0
x8:
y1 = 0.664570883960467
y2 = -0.22196013500854342
v01,v11,v21,..v02,v12,v22...w01,w11,...
-6.926315 0.548870 0.588708 0.279873 0.646394 0.255933 -0.017404 0.054164 0.083505 -0.202503 0.863752 -25.249966 0.525499 0.565337 0.256581 0.623023 0.232562 -0.040775 0.830793 0
Process finished with exit code 0
TODO Run Terminal Messages
Event Log
Build completed successfully in 4 s 280 ms (3 minutes ago)
4282:1 CRLF UTF-8 4 spaces
Type here to search
ENG 11:29 PM 7/18/2022