

Task 4
Student – Oksana Zanko
 $y = \sin^6(5\pi x)$

Results:

Code:

```
namespace Population
{
    class Chromosoma
    {
        private static int num_in_ch = 20;
        private static Random rnd = new Random();
        private double handsome = num_in_ch;
        //ceil
        private double[] a = new double[num_in_ch];

        public Chromosoma()
        {
            GenerateRandom();
        }

        public Chromosoma(Chromosoma _a, Chromosoma _b)
        {
            for (int i = 0; i < num_in_ch; i++)
            {
                //rnd.NextDouble(10)
                a[i] = rnd.Next(2) > 0 ? _a.getCeilByNum(i) : _b.getCeilByNum(i);

                handsome += Math.Pow(Math.Sin(5 * Math.PI * a[i]), 6);
            }
        }

        public double getCeilByNum(int i)
        {
            return a[i];
        }

        private void GenerateRandom()
        {
            for (int i = 0; i < num_in_ch; i++)
            {
                a[i] = rnd.Next(-5, 6) * rnd.NextDouble();
                handsome += Math.Pow(Math.Sin(5 * Math.PI * a[i]), 6);
            }
        }

        public double GetHandsome()
        {
            return handsome;
        }
    }

    class Population
    {
        private static int population_count = 300;
        private static Random rnd = new Random();
```

```

List<Chromosoma> population;

public Population()
{
    population = new List<Chromosoma>();
    for (int i = 0; i < population_count; i++)
        population.Add(new Chromosoma());
}

public void SortChromosoms()
{
    Chromosoma a;
    for (int i = 0; i < population_count; i++)
        for (int j = 0; j < population_count - 1; j++)
        {
            if (population[j].GetHandsome() > population[j + 1].GetHandsome())
            {
                a = population[j];
                population[j] = population[j + 1];
                population[j + 1] = a;
            }
        }
}

public void MakeItHalf()
{
    population.RemoveRange(population_count / 2, population_count / 2);
}

public void MakeChildren()
{
    int temp = population.Count;
    for (int i = 0; i < temp; i++)
    {
        int first = rnd.Next(0, population.Count);
        int second = rnd.Next(0, population.Count);
        population.Add(new Chromosoma(population[first], population[second]));
    }
}

public double GetAveregeHandsome()
{
    double handsome = 0;

    for (int i = 0; i < population.Count; i++)
        handsome += population[i].GetHandsome();

    handsome /= population.Count;

    return handsome;
}

public double GetTopHandsome()
{
    return population[0].GetHandsome();
}

public double GetHandSome(int i)
{
    return population[i].GetHandsome();
}

public bool Check()
{

```

```

        return population[0].GetHandsome() == 0;
    }
}

class Program
{
    static void Main(string[] args)
    {
        StreamWriter f = new StreamWriter(@"C:\AveregeChromosom.csv");
        StreamWriter f2 = new StreamWriter(@"C:\TopChromosom.csv");
        int generation = 1;
        Population a = new Population();
        f.WriteLine(a.GetAveregeHandsome() + "!" + generation);
        f2.WriteLine(a.GetTopHandsome() + "!" + generation);
        int count = 0;
        int same = 0;
        double last = 0;
        while (true)
        {
            if (last == a.GetTopHandsome())
                same++;
            else
                same = 0;
            last = a.GetTopHandsome();
            if (same == 5)
                break;
            generation++;
            a.SortChromosoms();
            a.MakeItHalf();
            a.MakeChildren();
            Console.WriteLine(a.GetTopHandsome());
            f.WriteLine(a.GetAveregeHandsome() + "!" + generation);
            f2.WriteLine(a.GetTopHandsome() + "!" + generation);
            if (a.Check())
                break;
            count++;
        }

        f.Close();
        f2.Close();
    }
}

```

```

#include <iostream>
#include <time.h>
#include <math.h>
#include <stdio.h>

```

```
using namespace std;
```

```

void main()
{
    int parents[20][10], selection[10][10], i, j, temp, number;
    int cros, Num1, Num2;
    float y[20], x[20], x1;

    srand(time(NULL));
    for (i=0; i<20; i++)
    {
        for (j=0; j<10; j++)
        {

```

```

        parents[i][j]=rand()%2;
    }
}

//fitness
for (i=0; i<20; i++)
{
    x[i]=0;
    for (j=0; j<10; j++)
    {
        x[i]+=parents[i][j]*pow(2.0,(9-j));
    }
    x[i]=x[i]/1023;
    y[i]=pow(sin(5*3.14*x[i]),6);
}

//sort
for (i=0; i<19; i++)
{
    if (y[i]>y[i+1])
    {
        x1=y[i];
        y[i]=y[i+1];
        y[i+1]=x1;
        for (j=0; j<10; j++)
        {
            temp=parents[i][j];
            parents[i][j]=parents[i+1][j];
            parents[i+1][j]=temp;
        }
        i=0;
    }
}

FILE *pfile;
pfile=fopen("text.txt","w");

for (i=0; i<20; i++)
{
    for (j=0; j<10; j++)
    {
        cout<<parents[i][j]<<" ";
    }
    cout<<endl;
}
cout<<endl<<endl;
number=0;

do
{
    for (i=0; i<20; i++)
    {
        fprintf(pfile,"%t%.10f\t%.10f\n",y[i],x[i]);
    }
    fprintf(pfile,"\n-----\n");
}

\n");

//selection
for (i=0; i<10; i++)
    for (j=0; j<10; j++)
        selection[i][j]=parents[i][j];

//crossover

```

```

        for (i=0; i<20; i++)
        {
            cros=rand()%8+1;
            Num1=rand()%10;
            Num2=rand()%10;
            for (j=0; j<10; j++)
            {
                if (j<cros)
                    parents[i][j]=selection[Num1][j];
                else
                    parents[i][j]=selection[Num2][j];
            }
        }

//fitness
for (i=0; i<20; i++)
{
    x[i]=0;
    for (j=0; j<10; j++)
    {
        x[i]+=parents[i][j]*pow(2.0, (9-j));
    }
    x[i]=x[i]/1023;
    y[i]=pow(sin(5*3.14*x[i]), 6);
}

//sort
for (i=0; i<19; i++)
{
    if (y[i]>y[i+1])
    {
        x1=y[i];
        y[i]=y[i+1];
        y[i+1]=x1;
        for (j=0; j<10; j++)
        {
            temp=parents[i][j];
            parents[i][j]=parents[i+1][j];
            parents[i+1][j]=temp;
        }
        i=0;
    }
}
number++;
//cout<<number;
}
//while(y[19]!=0.0);
while (number!=15);

cout<<endl<<endl;
cout<<"-----"<<endl;
cout<<"Number of iteration = "<<number<<endl;
cout<<"-----"<<endl;
fprintf(pfile, "\n-----\nNumber
of iteration = %d\n-----\n", number);

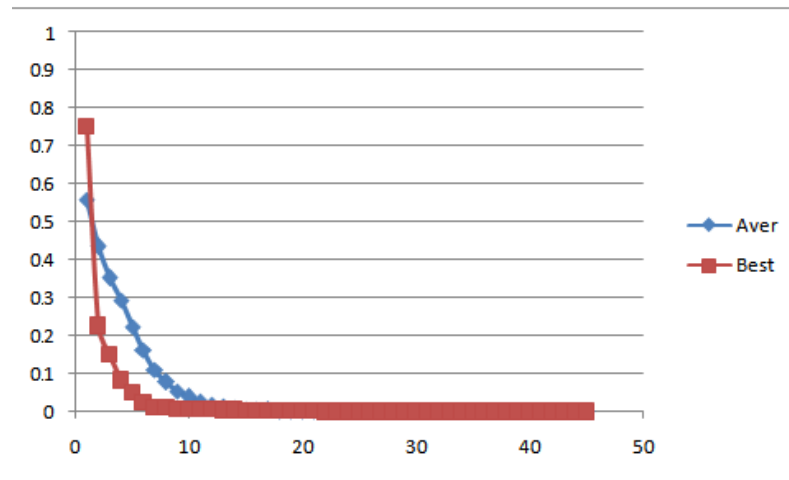
for (i=0; i<20; i++)
{
    for (j=0; j<10; j++)
    {
        cout<<parents[i][j]<<" ";
    }
    fprintf(pfile, "\t%.10f\t%.10f\n", y[i], x[i]);
    cout<<endl;
}

```

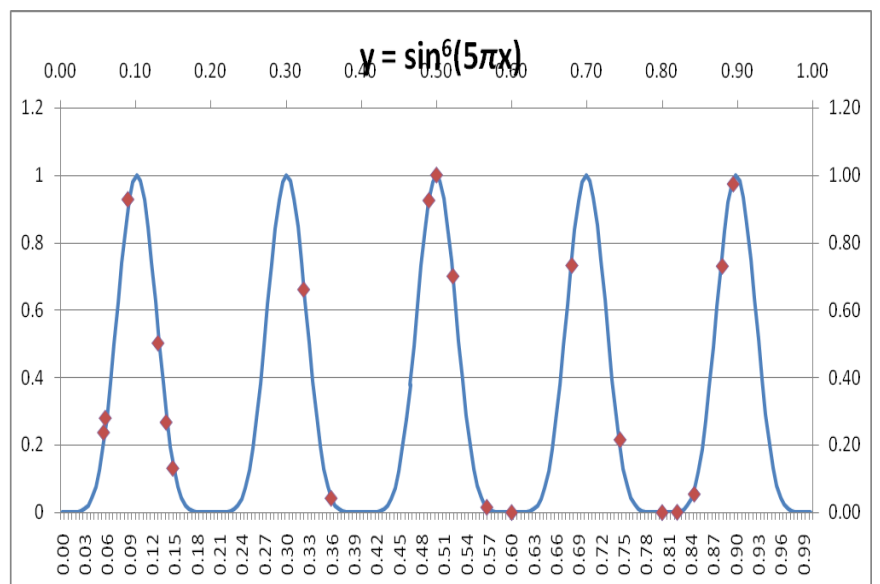
```

    }
    fclose(pfile);
    system("pause");
}

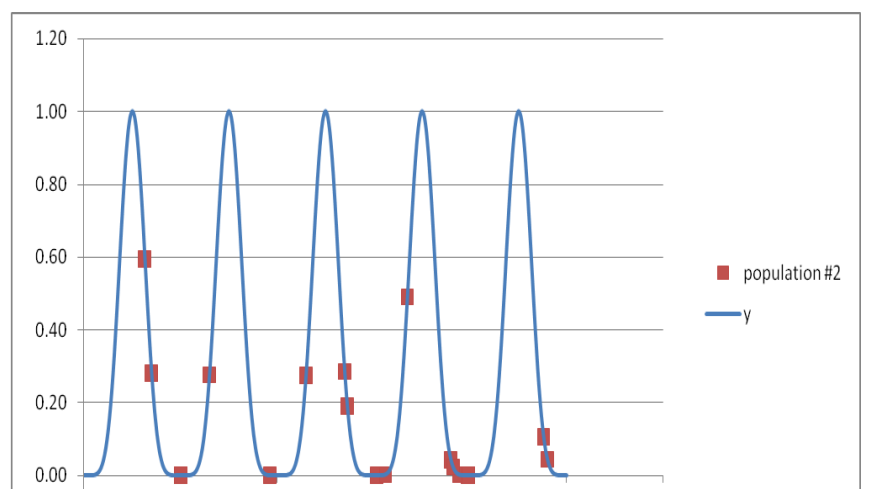
```



population #1	
0.73	0.88
0.92	0.49
0.00	0.60
0.22	0.74
0.05	0.84
0.00	0.82
0.70	0.52
0.04	0.36
0.24	0.06
0.73	0.68
0.02	0.57
0.27	0.14
0.97	0.89
0.13	0.15
0.66	0.32
0.50	0.13
0.00	0.80
1.00	0.50
0.28	0.06
0.93	0.09



population #2	
0.28	0.46
0.00	0.61
0.04	0.76
0.04	0.96
0.49	0.67
0.29	0.54



0.28	0.14
0.00	0.79
0.00	0.80
0.11	0.95
0.00	0.78
0.00	0.20
0.19	0.55
0.02	0.76
0.28	0.26
0.00	0.39
0.60	0.13
0.00	0.62
0.00	0.80
0.00	0.20