

Conditions

1. Create 100 random binary-chromosomes each with 1000 genes.
2. Fitness is the number of "1" in one chromosome – the more the better.
3. Select 2 chromosomes at random from the better half of the population.
4. Create a child chromosome by a one-point-crossover.
5. Give the child a mutation with a probability of $1/1000 = 0.001$.
6. Repeat from 2. to 5. 100 times and create the next generation.
7. Repeat 6. until the fitness value does not change any more.
8. Show the result:
 - (1) Display the best chromosome in the 1st, an intermediate & final generation.
 - (2) Display the best and average fitness vs. generation.

Source code

```
#include "stdafx.h"
#include "time.h"
#include <iostream>
using namespace std;

int generation[100][1000];
int fitness[100];
int gen = 0;
int prev_max;
int q=0;
void fit()
{
    memset(fitness,0,100*sizeof(int));
    for(int i=0;i<100;i++)
        for(int j=0;j<1000;j++)
            if(generation[i][j])
                fitness[i]++;
}

void sort()
{
    for(int i=0;i<100;i++)
        for(int j=0;j<100;j++)
            if(i!=j &&((i<j&&fitness[i]<fitness[j]) || (i>j&&fitness[i]>fitness[j])))
            {
                for(int k=0;k<1000;k++)
                {
                    int gen_t=generation[i][k];
                    generation[i][k]=generation[j][k];
```

```

                                generation[j][k]=gen_t;
                                }
                                int fit_t = fitness[i];
                                fitness[i]=fitness[j];
                                fitness[j]=fit_t;
                                }
}

void cross()
{
    int new_gener[100][1000];
    for(int i=0;i<50;i++)
    {
        int mama=rand()%50;
        int papa=rand()%50;
        int cros_gen=rand()%998+1;
        for(int k=0;k<cros_gen;k++)
            new_gener[i*2][k]=generation[mama][k];
        for(int j=cros_gen;j<1000;j++)
            new_gener[i*2][j]=generation[papa][j];

        for(int k=0;k<cros_gen;k++)
            new_gener[i*2+1][k]=generation[papa][k];
        for(int j=cros_gen;j<1000;j++)
            new_gener[i*2+1][j]=generation[mama][j];
    }
    memcpy(generation,new_gener,100*1000*sizeof(int));
}

void mutation()
{
    for(int i=0;i<100;i++)
        for(int j=0;j<1000;j++)
            if(rand()%1000==0)
            {
                generation[i][j]=1-generation[i][j];
            }
}

void stats()
{
    cout<<"max: "<<fitness[0]<<endl;
    int average=0;
    for(int i=0;i<100;i++)
        average+=fitness[i];
    average/=100;
    cout<<"aver: "<<average<<endl;
    cout<<"generation: "<<++gen<<endl;
    cout<<"_____ "<<endl;
    if(prev_max==fitness[0])
        q++;
    else
    {
        prev_max=fitness[0];
        q=0;
    }
}

```

```

    }
    if(q==15 || fitness[0]==1000)
    {
        if(q==15)
            cout<<"there isn't any changes in statistics for 15 generations."<<endl;
        else
            cout<<"All One problem solved."<<endl;
            exit(0);
    }

}

int _tmain(int argc, _TCHAR* argv[])
{
    srand(time(0));
    for(int i=0;i<100;i++)
        for(int j=0;j<1000;j++)
            generation[i][j]=rand()%2;
    for(int i=0;i<1000;i++)
    {
        fit();
        sort();
        stats();
        cross();
        mutation();
    }
    return 0;
}

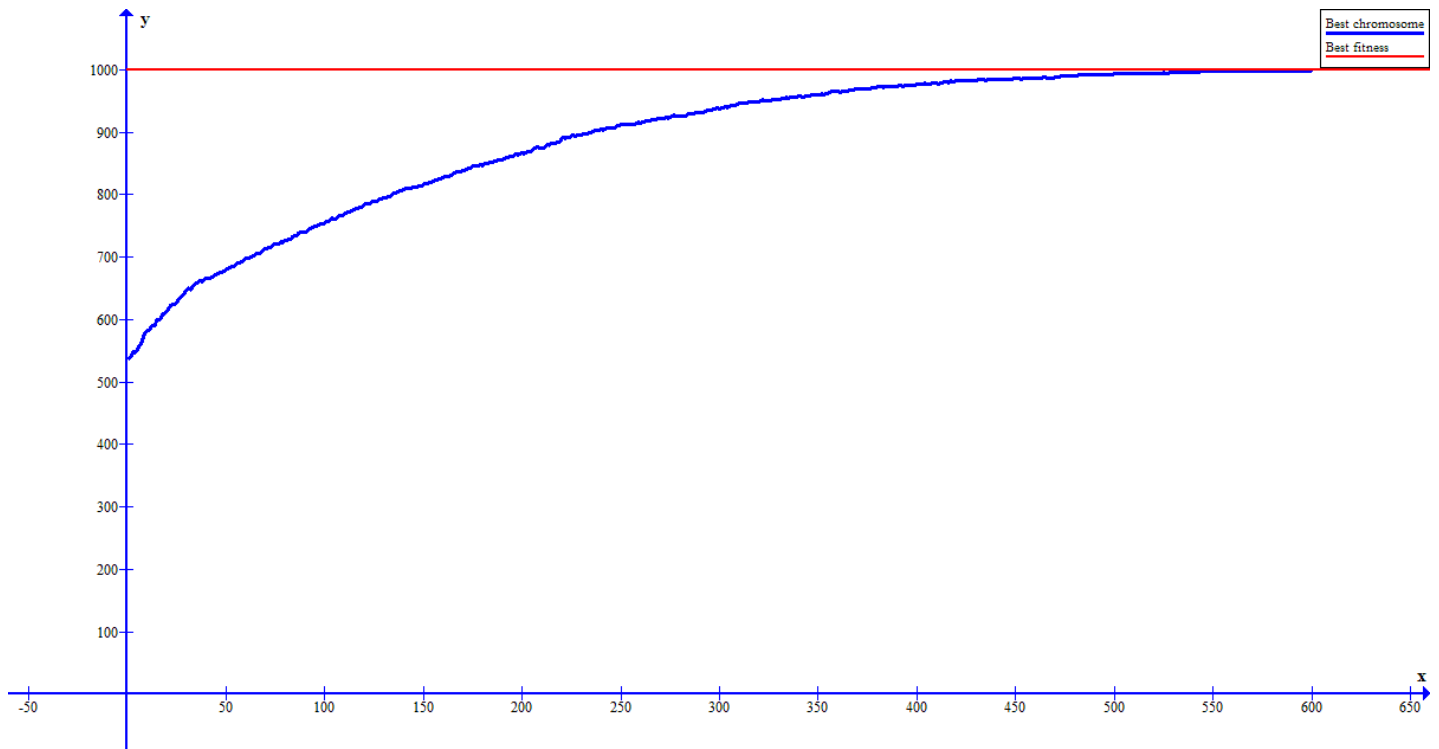
```

Results

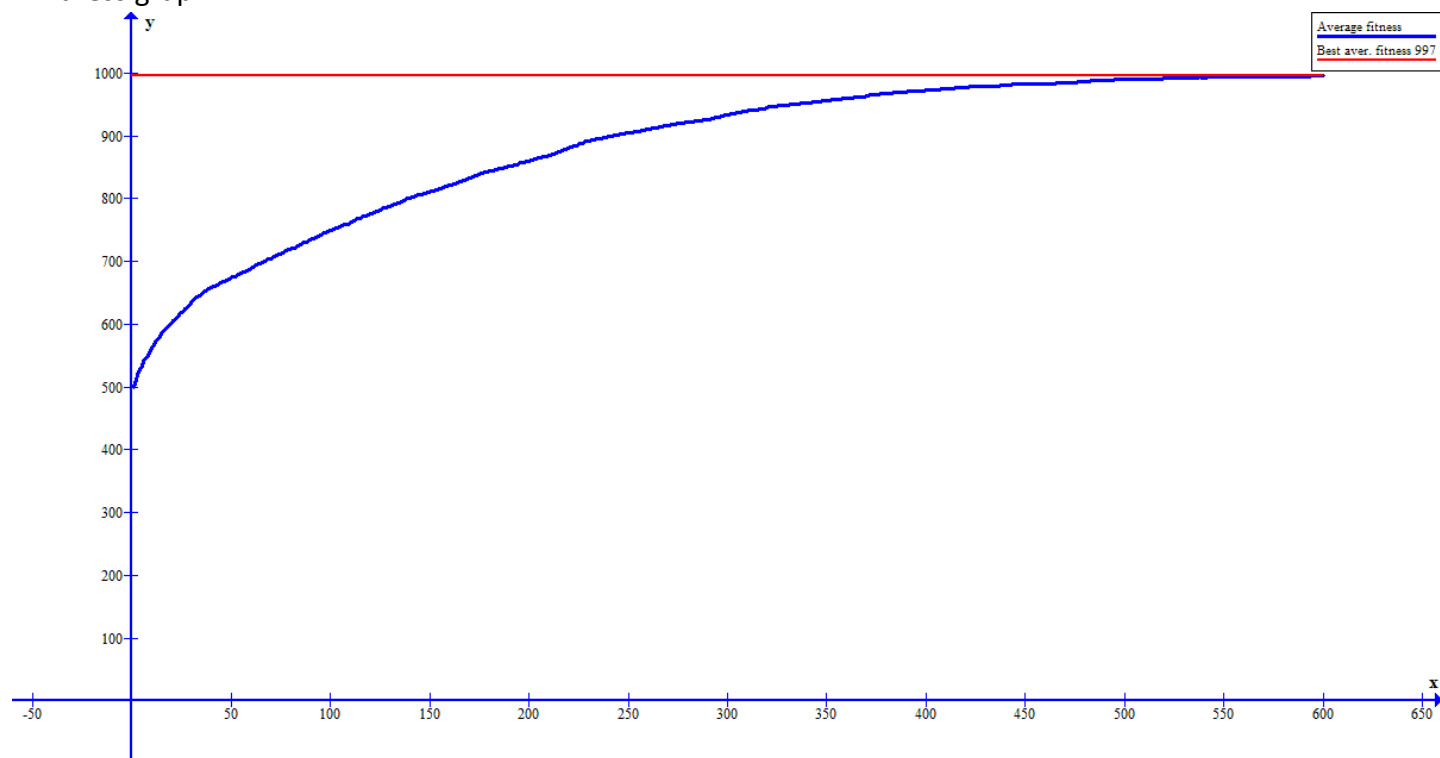
1. Best chromosome:

Start fitness: 537.

Best fitness: 1000.



2.Fitness graph:



Best average fitness: 997.

Starting average fitness: 499.

Generations: 600.