	fitness a	fitness b	domin	not domin
1010101110	0.00336491	0.0201618	0	12
10001100	0.0435114	0.166949	7	3
11111000	0.0777887	0.229351	10	1
110011101	0.105371	0.0155275	4	5
1000000011	0.123185	0.0227939	7	4
1111101100	0.0332065	0.146097	6	4
10001010	0.00698791	0.0804254	3	6
1111111110	0.00980564	0.0101963	0	12
1000110101	0.074448	0.00530735	0	11
100011011	0.198651	0.06037	11	1
1101101110	0.00407902	0.0185321	0	12
10011101	0.104107	0.0150446	3	6
11011000	0.0867487	0.244561	12	0
110001111	0.295239	0.117896	14	0
1000001011	0.170963	0.0455722	8	3
1111111000	0.0761631	0.226554	9	2
10001010	0.00698791	0.0804254	3	6
1111111010	0.000674782	0.0510654	0	10
1110101	0.0782251	0.0063501	1	8
11011	0.196914	0.0594141	10	2
1010101110	0.00336491	0.0201618	0	12
10001100	0.0435114	0.166949	7	3
11111000	0.0777887	0.229351	11	1
110011101	0.105371	0.0155275	4	5
1000000011	0.123185	0.0227939	7	4
1111101100	0.0332065	0.146097	6	4
10001010	0.00698791	0.0804254	3	6
1111111110	0.00980564	0.0101963	0	12
1000110101	0.074448	0.00530735	0	10
100011011	0.198651	0.06037	11	1
1101101110	0.00407902	0.0185321	0	12
10011101	0.104107	0.0150446	3	6
11011000	0.0867487	0.244561	12	0
110001111	0.295239	0.117896	14	0
1000001011	0.170963	0.0455722	8	3
1111111000	0.0761631	0.226554	10	2
10001010	0.00698791	0.0804254	3	6
1111111010	0.000674782	0.0510654	0	10
1000110101	0.074448	0.00530735	0	10
11011	0.196914	0.0594141	10	2
1010101110	0.00336491	0.0201618	0	12
10001100	0.0435114	0.166949	7	3
11111000	0.0777887	0.229351	11	1
110011101	0.105371	0.0155275	4	5
1000000011	0.123185	0.0227939	7	4

1111101100	0.0332065	0.146097	6	4
10001010	0.00698791	0.0804254	3	6
1111111110	0.00980564	0.0101963	0	12
1000110101	0.074448	0.00530735	0	10
100011011	0.198651	0.06037	11	1
1101101110	0.00407902	0.0185321	0	12
10011101	0.104107	0.0150446	3	6
11011000	0.0867487	0.244561	12	0
110001111	0.295239	0.117896	14	0
1000001011	0.170963	0.0455722	8	3
1111111000	0.0761631	0.226554	10	2
10001010	0.00698791	0.0804254	3	6
1111111010	0.000674782	0.0510654	0	10
1000110101	0.074448	0.00530735	0	10
11011	0.196914	0.0594141	10	2
1010101110	0.00336491	0.0201618	0	12
10001100	0.0435114	0.166949	7	3
11111000	0.0777887	0.229351	11	1
110011101	0.105371	0.0155275	4	5
1000000011	0.123185	0.0227939	7	4
1111101100	0.0332065	0.146097	6	4
10001010	0.00698791	0.0804254	3	6
1111111110	0.00980564	0.0101963	0	12
1000110101	0.074448	0.00530735	0	10
100011011	0.198651	0.06037	11	1
1101101110	0.00407902	0.0185321	0	12
10011101	0.104107	0.0150446	3	6
11011000	0.0867487	0.244561	12	0
110001111	0.295239	0.117896	14	0
1000001011	0.170963	0.0455722	8	3
1111111000	0.0761631	0.226554	10	2
10001010	0.00698791	0.0804254	3	6
1111111010	0.000674782	0.0510654	0	10
1000110101	0.074448	0.00530735	0	10
11011	0.196914	0.0594141	10	2
1010101110	0.00336491	0.0201618	0	12
10001100	0.0435114	0.166949	7	3
11111000	0.0777887	0.229351	11	1
110011101	0.105371	0.0155275	4	5
1000000011	0.123185	0.0227939	7	4
1111101100	0.0332065	0.146097	6	4
10001010	0.00698791	0.0804254	3	6
1111111110	0.00980564	0.0101963	0	12
1000110101	0.074448	0.00530735	0	10
100011011	0.198651	0.06037	11	1
1101101110	0.00407902	0.0185321	0	12

10011101	0.104107	0.0150446	3	6
11011000	0.0867487	0.244561	12	0
110001111	0.295239	0.117896	14	0
1000001011	0.170963	0.0455722	8	3
1111111000	0.0761631	0.226554	10	2
10001010	0.00698791	0.0804254	3	6
1111111010	0.000674782	0.0510654	0	10
1000110101	0.074448	0.00530735	0	10
11011	0.196914	0.0594141	10	2

```
#include <iostream>
#include <time.h>
#include <fstream>
using namespace std;
float fitnessa(bool b[10])
{
	float x =( b[9] * pow(2, 9) + b[8] * pow(2, 8) + b[7] * pow(2, 7) + b[6] * pow(2, 6) + b[5] * pow(2, 5) + b[4] * pow(2, 4) + b[3] * pow(2, 3) + b[2] * pow(2, 2) + b[1] * pow(2, 1) + b[0] * pow(2, 0))/1024;
	return pow((x - 0.4), 2);
}
float X(bool b[10])
{

	return (b[9] * pow(2, 9) + b[8] * pow(2, 8) + b[7] * pow(2, 7) + b[6] * pow(2, 6) + b[5] * pow(2, 5) + b[4] * pow(2, 4) + b[3] * pow(2, 3) + b[2] * pow(2, 2) + b[1] * pow(2, 1) + b[0] * pow(2, 0))/1024;
}
float fitnessb(bool b[10])
{
	float x = (b[9] * pow(2, 9) + b[8] * pow(2, 8) + b[7] * pow(2, 7) + b[6] * pow(2, 6) + b[5] * pow(2, 5) + b[4] * pow(2, 4) + b[3] * pow(2, 3) + b[2] * pow(2, 2) + b[1] * pow(2, 1) + b[0] * pow(2, 0)) / 1024;
	return pow((x - 0.6), 2);
}
float fitness(bool b[10])
{
	float x = (b[9] * pow(2, 9) + b[8] * pow(2, 8) + b[7] * pow(2, 7) + b[6] * pow(2, 6) + b[5] * pow(2, 5) + b[4] * pow(2, 4) + b[3] * pow(2, 3) + b[2] * pow(2, 2) + b[1] * pow(2, 1) + b[0] * pow(2, 0)) / 1024;
	return pow((x - 0.4), 2)-pow((x - 0.6), 2);
}
void main()
{
	srand(time(NULL));
	bool t[101][10], l[50][10];
	int i, y, k, kp = 0.8, tt, s, c = 0, cr,O=0,a,b;
	for (i = 0; i < 40; i++)
	for (y = 0; y < 10; y++)
		t[i][y] = rand() % 2;
	ofstream g=ofstream("E:\\tt.txt", ios::in|ios::out|ios::trunc);
	do
	{
		c++;
		for (i = 0; i < 20; i++)
		{
			for (y = 0; y < 10; y++)
				cout << (t[i][y] ? 1 : 0);
			cout << " " << fitnessa(t[i]) << " " << fitnessb(t[i]) << endl;
		}
		cout << "----------" << endl;
		for (i = 20; i < 30; i++)
		{
			tt = i - 20;
			s = i - 10;
			for (y = 0; y < 4; y++)
```

```
                        t[i][y] = t[tt][y];
                for (y = 4; y < 10; y++)
                        t[i][y] = t[s][y];
                if (fitnessa(t[tt])>fitnessa(t[i]) && fitnessb(t[tt])>fitnessb(t[i]))
                for (y = 0; y < 10; y++)
                        t[tt][y] = t[i][y];
                else
                if (fitnessa(t[s])>fitnessa(t[i]) && fitnessb(t[s]) > fitnessb(t[i]))
                for (y = 0; y < 10; y++)
                        t[s][y] = t[i][y];
        }

        for (int tt = 0; tt < 20; tt++){
                for (i = 0, a = 0, b = 0; i < 20; i++)
                {
                        if (fitnessa(t[tt])>fitnessa(t[i]) && fitnessb(t[tt])>fitnessb(t[i]))
                                a++;
                        if (fitnessa(t[tt]) < fitnessa(t[i]) && fitnessb(t[tt]) < fitnessb(t[i]))
                                b++;
                }
                g << fitnessa(t[tt]) << "\t" << fitnessb(t[tt]) << "\t" << a << "\t" << b << "\t" << X(t[tt]) << "\t"
<< pow(X(t[tt]) - 0.4, 2) << "\t"<<pow(X(t[tt]) - 0.6, 2)<< endl;
        }
    }
    while (c<5);
}
```