

```

using System;
using System.Linq;

namespace лаба_6
{
    class TableOfParam
    {
        public Random rnd = new Random();
        public int[] xPrice = new int [20];
        public int[] yVolume = new int [20];
        public TableOfParam()
        {
            for (int i = 0; i < 20; i++)
            {
                xPrice[i] = rnd.Next(1, 2);
                Console.Write(xPrice[i] + " ");
            }
            Console.WriteLine();
            for (int i = 0; i < 20; i++)
            {
                yVolume[i] = rnd.Next(1, 2);
                Console.Write(yVolume[i] + " ");
            }
        }
    }

    class Class1
    {
        TableOfParam tB = new TableOfParam();
        Random rnd = new Random();
        const int gen = 20;
        const int chromosome = 40;
        const int maxParam = 10;
        const int sizeOfBag = 2000;
        int[,] population = new int [chromosome, gen];
        int[,] newPopulation = new int[chromosome / 2, gen];
        int[] fitness = new int[chromosome];
        int[] keys = new int[chromosome];
        int[,] p = new int[chromosome, gen];
        public Class1()
        {
            //создание поколения
            int sumPrice;
            int sumVolume;
            for( int i = 0; i < chromosome; i++)
            {
                sumPrice = 0;
                sumVolume = 0;
                for (int j = 0; j < gen; j++)
                {
                    population[i, j] = rnd.Next(1, 101);
                    sumPrice += (tB.xPrice[j] * population[i, j]);
                    sumVolume += (tB.yVolume[j] * population[i, j]);
                }
                //считаем фитнесы первого поколения
                fitness[i] = (sumVolume > sizeOfBag) ? sumPrice : sumPrice;
            }
        }

        public void Cross()
    }
}

```

```

{
    //сортировка
    for (int i = 0; i < chromosome; i++)
    {
        keys[i] = i;
    }
    Array.Sort(fitness, keys);
    for (int i = 0, n = chromosome - 1; i < chromosome / 2; i++, n--)
    {
        for (int j = 0; j < gen; j++)
        {
            newPopulation[i, j] = population[keys[n], j];
        }
    }
    //кроссинговер
    int k = 0;
    for (int i = 0; i < chromosome / 2; i++)
    {
        int a = rnd.Next(0, chromosome / 2);
        int b = rnd.Next(0, chromosome / 2);
        for (int j = 0; j < gen; j++)
        {
            {
                p[0, j] = newPopulation[a, j];
                p[1, j] = newPopulation[b, j];
            }
            int c = rnd.Next(0, 2);
            for (int j = 0; j < gen; j++)
            {
                if (c == 0)
                {
                    {
                        population[k, j] = p[0, j];
                    }
                    else
                    {
                        {
                            population[k + 1, j] = p[1, j];
                        }
                    }
                }
            }
            k += 2;
        }
    }
}

public void Mutation()
{
    int sumPrice;
    int sumVolume;
    for (int i = 0; i < chromosome; i++)
    {
        sumPrice = 0;
        sumVolume = 0;
        for (int j = 0; j < gen; j++)
        {
            int value = rnd.Next(0, gen);
            population[i, j] = (value == 0) ? rnd.Next(1, 101) : population[i, j];
            sumPrice += (tB.xPrice[j] * population[i, j]);
            sumVolume += (tB.yVolume[j] * population[i, j]);
        }
        fitness[i] = (sumVolume > sizeOfBag) ? sumPrice : sumPrice;
    }
}

public int GetMax()

```

```

    {
        return fitness.Max();
    }
    public int GetAverage()
    {
        return (int)fitness.Average();
    }
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

namespace лаба_6

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            Class1 cl = new Class1();
            int i = -1;
            cl.Cross();
            cl.Mutation();
            Console.WriteLine("*****");
            while (true)
            {
                i++;
                cl.Cross();
                cl.Mutation();
                Console.WriteLine(cl.GetMax() + " ");
            }
        }
    }
}
}

```