

```
// 34, задача 1
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Задача_1
{
    class Program
    {
        static Int64 insert(string a)
        {
            Int64 n = 0;
            bool ok = true;
            while (ok)
            {
                ok = false;
                Console.Write(a);
                try
                {
                    n = Int64.Parse(Console.ReadLine());
                }
                catch
                {
                    Console.Write("Вы ввели что-то неправильно. ");
                    ok = true;
                }
            }
            return n;
        }

        static Int64 a1, b1, c1, a2, b2, c2;

        static Int64 bezizlishkov(Int64 n) //покупка без излишков
        {
            Int64 sum = (n / 144) * 11400;
            a1 = n / 144;
            n %= 144;
            sum += (n / 12) * 1024;
            b1 = n / 12;
            n %= 12;
            sum += n * 105;
            c1 = n;
            return sum;
        }

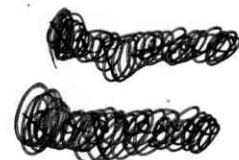
        static Int64 sizlishkami(Int64 n) //покупка с излишками
        {
            Int64 a21, a22;
            Int64 sum = (n / 144 + 1) * 11400;
            a21 = n / 144 + 1;
            b2 = 0;
            c2 = 0;
            Int64 sum2 = (n / 144) * 11400;
            a22 = n / 144;
            sum2 += (n / 12 + 1) * 1024;
            b2 = n / 12 + 1;
            if(sum < sum2)
            {
                a2 = a21;
                b2 = 0;
                return sum;
            }
            else
            {
                a2 = a22;
                return sum2;
            }
        }
    }

    static void Main()

```

Lucas 1/2

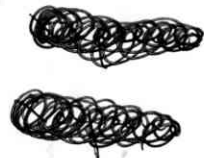
05.02.2016



```
{
    bool dow = true;
    while (dow)
    {
        Int64 n = insert("Введите количество пар: ");
        Int64 sum1 = bezizlishkov(n), sum2 = sizlishkami(n);
        if(sum1 < sum2)
        {
            Console.WriteLine("Оптимальной покупка будет, если взять без излишков.\nВыходит - {0}руб.\nНадо взять:\n{1} пар носков\n{2} связок носков\n{3} Коробок носков\nВсего выйдет {4} пар носков", sum1,
            c1, b1, a1, a1 * 144 + b1 * 12 + c1);
        }
        else
        {
            Console.WriteLine("Оптимальной покупка будет, если взять с излишками.\nЕсли брать без
            излишков, то выйдет {0}руб.\nC излишками - {1}руб.\nРазница - {2}руб.\nНадо взять:\n{3} пар носков\n{4}
            связок носков\n{5} Коробок носков\nВсего выйдет {6} пар носков", sum1, sum2, sum1 - sum2, c2, b2, a2, a2
            * 144 + b2 * 12 + c2);
        }

        Console.WriteLine("\n\nЧтобы Запустить еще раз, нажмите на пробел");
        int a = Console.ReadKey(true).KeyChar;
        if (a != 32)
        {
            dow = false;
        }
        Console.Clear();
    }
}
}
```

Лист 2/2
05.02.2016



// 34, задача 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Задача2
{
    class Program
    {
        static double[,] A;
        static int[][] B;
        static int[] V;
        static double sr;

        static void Inserter()// Заполнение матрицы
        {
            Random rnd = new Random();
            for (int i = 0; i < A.GetLength(0); i++)
            {
                for (int j = 0; j < A.GetLength(1); j++)
                {
                    int t = rnd.Next() % 2;
                    if (t == 0)
                        t = -1;
                    A[i, j] = (rnd.Next() % 10) * t;
                }
            }
        }

        static void OutA()// Вывод матрицы
        {
            Console.WriteLine("\nИсходная матрица A:");
            for (int i = 0; i < A.GetLength(0); i++)
            {
                for (int j = 0; j < A.GetLength(1); j++)
                {
                    string t = "";
                    if (A[i, j] >= 0)
                        t = " ";
                    Console.Write("{0}{1} ", t, A[i, j]);
                }
                Console.WriteLine();
            }
        }

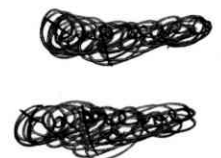
        static void FindSr() //Нахождение среднеарифмитического
        {
            Int64 us = 0, sum = 0, dou = (A.GetLength(0) + 1) / 2, dof = 1;

            for (int i = 0; i < A.GetLength(0); i++)
            {
                dof = i + 1;
                if(dof > dou)
                {
                    dof = A.GetLength(0) - dof + 1;
                }
                for (int j = 0; j < dof; j++, us++)
                {
                    sum += (Int64)A[i, A.GetLength(1) - 1 - j];
                }
                if (dof < dou)
                    dof++;
                else
                    dof--;
            }
            sr = sum / (double)us;
        }

        static void MakerB() // Создание ступенчатого массива
        {
```

Лисин 1/3

05.02.2016



```
for (int i = 0; i < A.GetLength(0); i++) // Вывод матрицы
{
    int m = 0;
    for (int j = 0; j < A.GetLength(1); j++)
    {
        if (A[i, j] > sr)
            m++;
    }
    B[i] = new int[m];
    for (int j = 0, added = 0; j < A.GetLength(1); j++)
    {
        if (A[i, j] > sr)
        {
            B[i][added] = (int)A[i, j];
            added++;
        }
    }
}

static void OutB() // Вывод ступенчатого массива
{
    Console.WriteLine("\nСтупенчатый массив B:");
    for (int i = 0; i < B.GetLength(0); i++)
    {
        for (int j = 0; j < B[i].Count(); j++)
        {
            string t = "";
            if (B[i][j] >= 0)
                t = " ";
            Console.Write("{0}{1} ", t, B[i][j]);
        }
        if (B[i].Count() == 0)
            Console.WriteLine("nr");
        Console.WriteLine();
    }
}

static void MakerV() // Создание массива с минимальными элементами
{
    for (int i = 0; i < B.GetLength(0); i++)
    {
        if (B[i].Count() == 0)
        {
            V[i] = -1000;
            continue;
        }
        int min = B[i][0];
        for (int j = 1; j < B[i].Count(); j++)
        {
            if (B[i][j] < min)
                min = B[i][j];
        }
        V[i] = min;
    }
}

static void OutV() // Вывод массива с минимальными элементами
{
    Console.WriteLine("\nМинимальные элементы строк:");
    for (int i = 0; i < V.Count(); i++)
    {
        if (V[i] == -1000)
            Console.WriteLine("nr ");
        else
            Console.WriteLine("{0} ", V[i]);
    }
}

static void Main()
{
    bool dow = true;
    while (dow)
    {
```

Lucas 2/3

05.02.2016



```
Int64 n = 0;
bool ok = true;
while (ok)
{
    ok = false;
    Console.Write("Введите размер матрицы: ");
    try
    {
        n = Int64.Parse(Console.ReadLine());
    }
    catch
    {
        Console.Write("Вы ввели что-то неправильно. ");
        ok = true;
    }
}

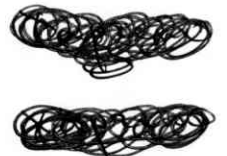
A = new double[n, n];
B = new int[n][];
V = new int[n];

Inserter(); // Заполнение матрицы
OutA(); // Вывод матрицы
FindSr(); //Нахождение среднеарифмитического
Console.WriteLine("\nСреднее = {0:F2}", sr);
MakerB(); // Создание ступенчатого массива
OutB(); // Вывод ступенчатого массива
MakerV(); // Создание массива с минимальными элементами
OutV(); // Вывод смассива с минимальными элементами

Console.WriteLine("\n\nЧтобы Запустить еще раз, нажмите на пробел");
int a = Console.ReadKey(true).KeyChar;
if(a != 32)
{
    dow = false;
}
Console.Clear();
}
}
}
```

Лист 3/3

05.02.2016



// 34, задача 3

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Задача_3
{
    class Program
    {
        static bool[] was;
        static int[] A;
        static int M;

        static int insert(string a)
        {
            int n = 0;
            bool ok = true;
            while (ok)
            {
                ok = false;
                Console.Write(a);
                try
                {
                    n = int.Parse(Console.ReadLine());
                }
                catch
                {
                    Console.Write("Вы ввели что-то неправильно. ");
                    ok = true;
                }
            }
            return n;
        }

        static void sol(int i, int sum)
        {
            was[i] = true;
            sum += A[i];
            if(sum == M)
            {
                Console.WriteLine("Сумму {0} можно разменять следующими купюрами: ", M);
                for (int j = 0; j < A.Count(); j++)
                {
                    if (was[j])
                    {
                        Console.WriteLine("{0} ", A[j]);
                    }
                }
                Console.WriteLine();
                was[i] = false;
                return;
            }
            if(sum > M)
            {
                was[i] = false;
                return;
            }
            for (int j = i + 1; j < was.Count(); j++)
            {
                sol(j, sum);
            }
            was[i] = false;
        }

        static void Main()
        {
            bool dow = true;
            while (dow)
            {
                int n;
                n = insert("Введите количество купюр: ");
            }
        }
    }
}
```

Лусин 2.12

05.02.2016



```
M = insert("Введите сумму: ");
A = new int[n];
was = new bool[n];
for (int i = 0; i < n; i++)
{
    A[i] = insert("Введите достоинство купюры №" + (i + 1) + ": ");
}
for (int i = 0; i < n; i++)
{
    sol(i, 0);
}
}
```

```
Console.WriteLine("\n\nЧтобы Запустить еще раз, нажмите на пробел");
int a = Console.ReadKey(true).KeyChar;
if (a != 32)
{
    dow = false;
}
Console.Clear();
}
```

Лусин 2/2

05.02.2016



задача3

#33, задача 2

```
from copy import *

def MyInput():
    valid = False
    while not valid: # Проверка входных данных
        s = input()
        try:
            float(s)
        except ValueError:
            print('> Введено не число!')
            continue
        N = float(s)
        if N % 1 == 0:
            if N > 0:
                valid = True
            else:
                print('> Введено неположительное число!')
        else:
            print('> Введено нецелое число!')
    return int(s)

def GetWays(money, A, used = [], index = 0): # Рекурсивная функция, находящая
способы размена
    if money == 0:
        print(*used) # Вывод найденного способа
    elif len(used) < len(A):
        for i in range(index, len(A)): # Перебор вариантов
            new = copy(used)
            new.append(A[i])
            GetWays(money - A[i], A, new, i + 1)

want = True
while want:
    # Ввод данных
    print('> Введите сумму, которую нужно разменять')
    money = MyInput()
    print('> Введите количество купюр')
    N = MyInput()
    print('> Введите достоинства купюр по 1 на каждой строке')
    A = [MyInput() for i in range(N)]

    print('> Варианты размена:')

    GetWays(money, A) # Запуск рекурсивной функции

    print('> Введите 1, чтобы запустить заново, введите что-либо другое, чтобы
выйти.') # Перезапуск программы при желании
    answer = input()
    if answer != '1':
```


~~_____~~

Лист 2/2

want = False

задача3

~~_____~~

мет 1/2

#33, задача 1

```
want = True
while want:
    print('> Введите количество носков, которое нужно купить')
    valid = False
    while not valid: # Проверка входных данных
        N = input()
        try:
            float(N)
        except ValueError:
            print('> Введено не число!')
            continue
        N = float(N)
        if N % 1 == 0:
            if N > 0:
                valid = True
            else:
                print('> Введено неположительное число!')
        else:
            print('> Введено нецелое число!')

N = int(N)

box = N // (12 ** 2) # Подсчёт цены без оптимизации
pack = (N % (12 ** 2)) // 12
sock = N % 12
first_cost = box * 11400 + pack * 1025 + sock * 105

print('> Покупка без оптимизации:')
print('> Надо купить: коробок -', box, 'связок -', pack, 'отдельных пар носков -', sock)
print('> Цена:', first_cost)
print()

sock_price = sock * 105 # Оптимизация
if 1025 < sock_price:
    sock = 0
    pack += 1

variant = pack * 1025 + sock * 105

if 11400 < variant:
    box += 1
    pack = 0
    sock = 0
second_cost = box * 11400 + pack * 1025 + sock * 105

print('> Покупка с оптимизацией:')
print('> Надо купить: Коробок -', box, 'связок -', pack, 'отдельных пар носков -', sock)
print('> Цена:', second_cost, 'Выгода:', first_cost - second_cost)
```



задача1



```
print('> Введите 1, чтобы запустить заново, введите что-либо другое, чтобы  
выйти.')  
answer = input()  
if answer != '1':  
    want = False
```

#33, задача 2

```
from random import * # Подключаем библиотеку для получения случайных чисел
```

```
want = True
```

```
while want:
```

```
    valid = False
```

```
    print('> Введите размер матрицы')
```

```
    while not valid: # Проверка входных данных
```

```
        N = input()
```

```
        try:
```

```
            float(N)
```

```
        except ValueError:
```

```
            print('> Введено не число!')
```

```
            continue
```

```
        N = float(N)
```

```
        if N % 1 == 0:
```

```
            if N > 0:
```

```
                valid = True
```

```
            else:
```

```
                print('> Введено неположительное число!')
```

```
        else:
```

```
            print('> Введено нецелое число!')
```

```
N = int(N)
```

```
A = [[randint(0, 9) for i in range(N)] for i in range(N)] # Генерируем матрицу A
```

```
print('> Матрица A:')
```

```
for line in A:
```

```
    print(*line)
```

```
print()
```

```
height = N # Подсчитываем среднее арифметическое чисел в треугольнике
```

```
cur = -1
```

```
sum = 0
```

```
amount = 0
```

```
while height > 0:
```

```
    for i in range((N - height) // 2, N - ((N - height) // 2)):
```

```
        sum += A[i][cur]
```

```
        amount += 1
```

```
    height -= 2
```

```
    cur -= 1
```

```
average = sum / amount
```

```
print('> Среднее арифметическое чисел в треугольнике:', average)
```

```
print()
```

```
B = [[] for i in range(N)] # создаём массив B
```

```
for i in range(N):
```

```
    for j in range(N):
```

задача2

```
if A[i][j] > average:
    B[i].append(A[i][j])
if len(B[i]) == 0:
    B[i].append('nr')

print('> Массив B:')
for line in B:
    print(*line)
print()

V = [min(B[i]) for i in range(N)] # Создаём вектор V

print('> Вектор V:')
print(*V)

print('> Введите 1, чтобы запустить заново, введите что-либо другое, чтобы
выйти.')
answer = input()
if answer != '1':
    want = False
```