



Федеральное государственное бюджетное
образовательное учреждение
высшего образования
«Национальный исследовательский
университет «МЭИ»

ИНСТИТУТ ДИСТАНЦИОННОГО И ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ

Задание по учебной дисциплине: «Программирование»

Тема: «Динамические массивы.»

Вариант №6

Работу выполнил:

Студент группы ИДзс-21-22

Минин Дмитрий Михайлович

Москва 2023

Оглавление

Задача №2:.....	3
Условие	3
Состав данных	3
Блок-схема алгоритма	4
Программа на С++	6
Пример выполнения программы:	7
Задача №3:.....	8
Условие	8
Состав данных	8
Блок-схема алгоритма	9
Программа на С++	10
Пример выполнения программы:	11
Задача №4:.....	12
Условие	12
Состав данных	12
Блок-схема алгоритма	13
Программа на С++	14
Пример выполнения программы:	16

Задача №2:

Условие

1. Найти число n_1 отрицательных элементов вектора X и их сумму C_1 и число n_2 положительных элементов вектора Y и их сумму C_2 .

Состав данных

Имя	Назначение	Тип	Структура
<u>Исходные данные</u>			
n	Количество элементов вектора X и Y	целый	простая переменная
X	Вектор X	массив	одномерный
Y	Вектор Y	массив	одномерный
<u>Выходные данные</u>			
n1	Количество отрицательных элементов вектора X	целый	простая переменная
n2	Количество положительных элементов вектора Y	целый	простая переменная
C1	Сумма отрицательных элементов вектора X	целый	простая переменная
C2	Сумма положительных элементов вектора Y	целый	простая переменная
<u>Промежуточные данные</u>			
i	счётчик цикла	целый	простая переменная

Блок-схема алгоритма



Программа на C++

```
#include <iostream>
```

```
#include <stdlib.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n, n1 = 0, n2 = 0, C1 = 0, C2 = 0;
```

```
    cout << "Enter the number of vector elements: " << endl;
```

```
    cin >> n;
```

```
    int* X = new int[n];
```

```
    int* Y = new int[n];
```

```
    // cout << "Enter the elements of the vector X: " << endl;
```

```
    // Для удобства заполним массив псевдорандомными числами
```

```
    cout << "Vector X: " << endl;
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        X[i] = rand() % 100 - 50;
```

```
        cout << X[i] << endl;
```

```
        if (X[i] < 0)
```

```
        {
```

```
            n1++;
```

```
            C1 = C1 + X[i];
```

```
        }
```

```
    }
```

```
    cout << "Vector Y: " << endl;
```

```
    // cout << "Enter the elements of the vector Y: " << endl;
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        Y[i] = rand() % 100 - 50;
```

```
        cout << Y[i] << endl;
```

```
        if (Y[i] > 0)
```

```
        {
```

```
            n2++;
```

```
            C2 = C2 + Y[i];
```

```
        }
```

```
    }
```

```
    cout << "n1 = " << n1 << endl;
```

```
    cout << "n2 = " << n2 << endl;
```

```
    cout << "C1 = " << C1 << endl;
```

```
    cout << "C2 = " << C2 << endl;
```

```
    delete[] X, Y;
```

```
}
```

Пример выполнения программы:

```
Enter the number of vector elements:  
5  
Vector X:  
-9  
17  
-16  
-50  
19  
Vector Y:  
-26  
28  
8  
12  
14  
n1 = 3  
n2 = 4  
C1 = -75  
C2 = 62
```

Задача №3:

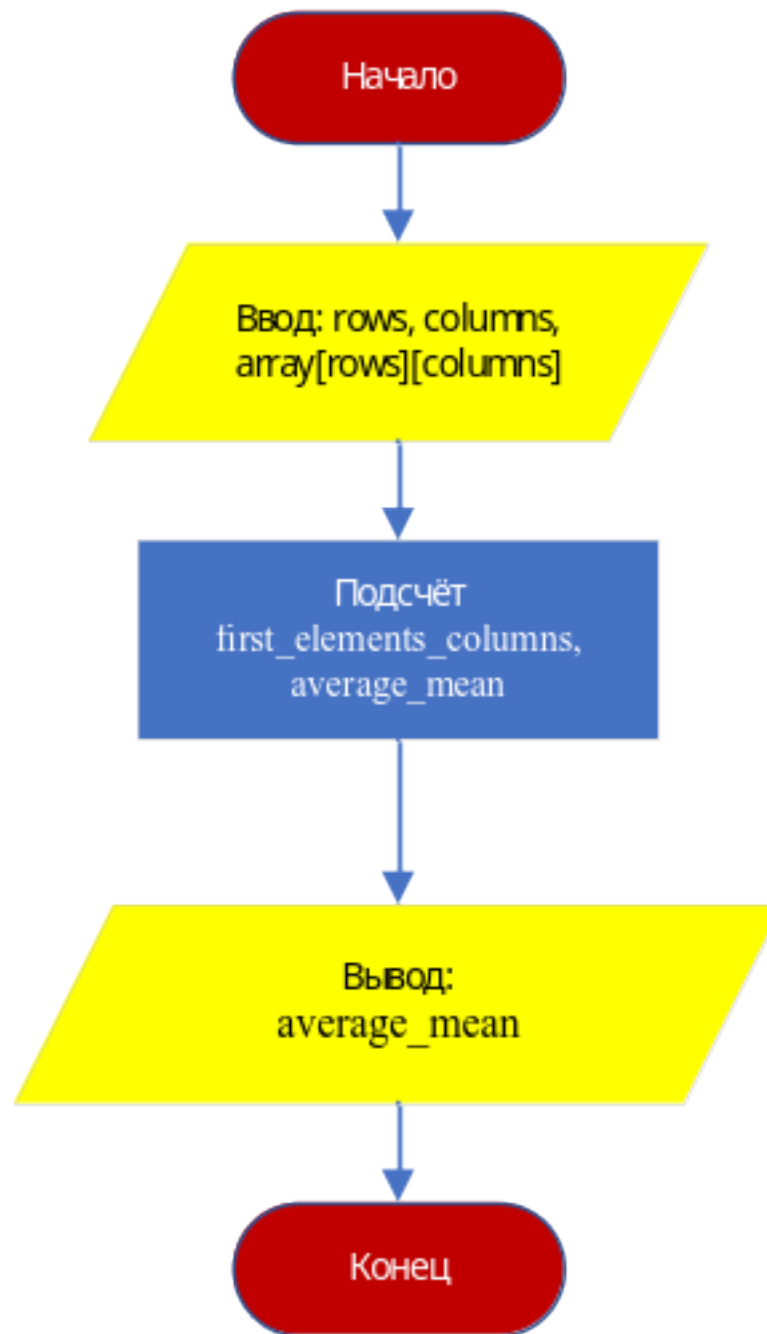
Условие

1. Для каждого столбца матрицы вычислить среднее арифметическое элементов, меньших первого элемента этого столбца

Состав данных

Имя	Назначение	Тип	Структура
<u>Исходные данные</u>			
rows	Количество строк матрицы	целый	простая переменная
columns	Количество столбцов матрицы	целый	простая переменная
array[rows][columns]	Инициализируем матрицу	массив	двумерный
<u>Выходные данные</u>			
first_elements_columns	Значения первых элементов столбцов	массив	одномерный
average_mean	Средние арифметические значения элементов столбцов, меньших первого элемента этого столбца	массив	одномерный
count	Количество элементов для подсчёта среднего арифметического значения	целый	простая переменная
<u>Промежуточные данные</u>			
i	счётчик цикла	целый	простая переменная
j	счётчик цикла	целый	простая переменная

Блок-схема алгоритма



Программа на C++

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "RUS");
    int rows, columns, counter;
    cout << "Введите количество строк матрицы: " << endl;
    cin >> rows;
    cout << "Введите количество столбцов матрицы: " << endl;
    cin >> columns;
    int* first_elements_columns = new int[columns];
    float* average_mean = new float[columns];
    int** array = new int* [rows];

    for (int i = 0; i < rows; i++)
    {
        array[i] = new int[columns];
    }

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            array[i][j] = rand() % 100;
        }
    }
    cout << endl;
    cout << "Матрица: " << endl;

    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            cout << array[i][j] << "\t";
        }
        cout << endl;
    }
    cout << endl;

    for (int i = 0; i < columns; i++)
    {
        first_elements_columns[i] = array[0][i];
    }

    cout << "Среднее арифметическое значений, что меньше первого элемента столбца:" <<
endl;

    for (int i = 0; i < columns; i++)
    {
        average_mean[i] = 0;
        counter = 0;
        for (int j = 1; j < rows; j++)
        {
```

```

        if (array[j][i] < first_elements_columns[i])
        {
            average_mean[i] += array[j][i];
            counter++;
        }
    }
    average_mean[i] = average_mean[i] / counter;
    cout << average_mean[i] << "\t";
}
cout << endl;

for (int i = 0; i < rows; i++)
{
    delete[] array[i];
}
delete[] array, first_elements_columns, average_mean;
}

```

Пример выполнения программы:

```

Введите количество строк матрицы:
7
Введите количество столбцов матрицы:
3
Матрица:
41      67      34
0       69      24
78      58      62
64      5       45
81      27      61
91      95      42
27      36      91

Среднее арифметическое значений, что меньше первого элемента столбца:
13.5    31.5    24

```

Задача №4:

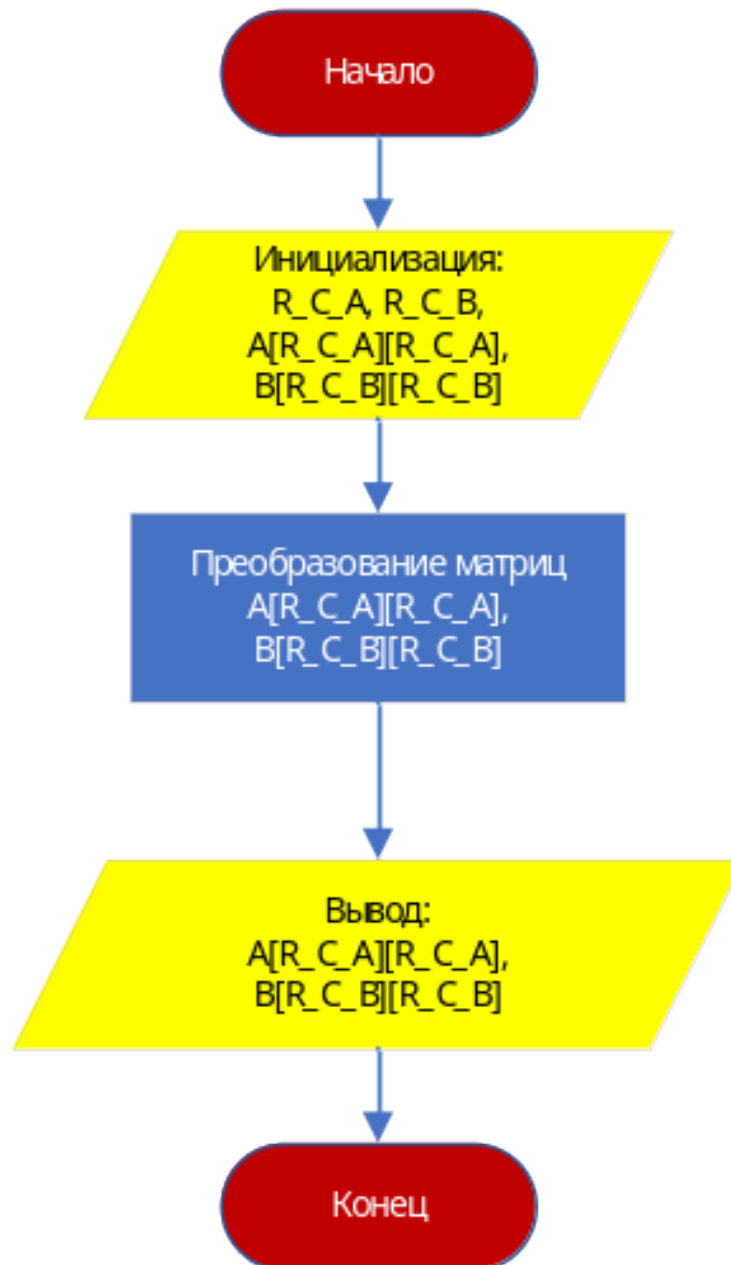
Условие

1. В матрицах A (7×7) и B (5×5) заменить каждый положительный элемент, стоящий выше главной диагонали, на нуль.

Состав данных

Имя	Назначение	Тип	Структура
<u>Исходные данные</u>			
R_C_A	Количество строк и колонок матрицы A	целый	простая переменная
R_C_B	Количество строк и колонок матрицы B	целый	простая переменная
A	Инициализируем матрицу	массив	двумерный
B	Инициализируем матрицу	массив	двумерный
<u>Выходные данные</u>			
A[R_C_A][R_C_A]	Преобразованная матрица	массив	двумерный
B[R_C_B][R_C_B]	Преобразованная матрица	массив	двумерный
<u>Промежуточные данные</u>			
i	счётчик цикла	целый	простая переменная
j	счётчик цикла	целый	простая переменная

Блок-схема алгоритма



Программа на C++

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "ru");
    const int R_C_A = 7;
    const int R_C_B = 5;
    int A[R_C_A][R_C_A], B[R_C_B][R_C_B];

    cout << "Матрица A:" << endl;
    for (int i = 0; i < R_C_A; i++)
    {
        for (int j = 0; j < R_C_A; j++)
        {
            A[i][j] = rand() % 100 - 50;
            if (A[i][j] > 0)
            {
                cout << " ";
            }
            cout << A[i][j] << "\t";
        }
        cout << endl;
    }
    cout << endl;

    cout << "Матрица B:" << endl;
    for (int i = 0; i < R_C_B; i++)
    {
        for (int j = 0; j < R_C_B; j++)
        {
            B[i][j] = rand() % 100 - 50;
            if (B[i][j] > 0)
            {
                cout << " ";
            }
            cout << B[i][j] << "\t";
        }
        cout << endl;
    }
    cout << endl;

    for (int i = 0; i < R_C_A; i++)
    {
        for (int j = 0; j < R_C_A; j++)
        {
            if (i < j)
            {
                if (A[i][j] > 0)
                {
```

```

        A[i][j] = 0;
    }
}
}

for (int i = 0; i < R_C_B; i++)
{
    for (int j = 0; j < R_C_B; j++)
    {
        if (i < j)
        {
            if (B[i][j] > 0)
            {
                B[i][j] = 0;
            }
        }
    }
}

cout << "Преобразованная матрица A:" << endl;
for (int i = 0; i < R_C_A; i++)
{
    for (int j = 0; j < R_C_A; j++)
    {
        if (A[i][j] > 0)
        {
            cout << " ";
        }
        cout << A[i][j] << "\t";
    }
    cout << endl;
}
cout << endl;

cout << "Преобразованная матрица B:" << endl;
for (int i = 0; i < R_C_B; i++)
{
    for (int j = 0; j < R_C_B; j++)
    {
        if (B[i][j] > 0)
        {
            cout << " ";
        }
        cout << B[i][j] << "\t";
    }
    cout << endl;
}
cout << endl;
}

```


Пример выполнения программы:

Матрица A:

-9	17	-16	-50	19	-26	28
8	12	14	-45	-5	31	-23
11	41	45	-8	-23	-14	41
-46	-48	3	42	32	-29	-34
-32	45	-3	-24	21	-12	19
-38	17	49	-15	44	-47	-39
-28	-17	23	14	-9	-39	3

Матрица B:

18	-3	-6	12	7
-13	9	-27	-9	-21
28	-34	-15	40	-8
38	-44	-10	-8	14
-2	-4	-45	40	-21

Преобразованная матрица A:

-9	0	-16	-50	0	-26	0
8	12	0	-45	-5	0	-23
11	41	45	-8	-23	-14	0
-46	-48	3	42	0	-29	-34
-32	45	-3	-24	21	-12	0
-38	17	49	-15	44	-47	-39
-28	-17	23	14	-9	-39	3

Преобразованная матрица B:

18	-3	-6	0	0
-13	9	-27	-9	-21
28	-34	-15	0	-8
38	-44	-10	-8	0
-2	-4	-45	40	-21