



**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет**

**«СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

**Институт  
информационных  
технологий**

**Кафедра  
информационных систем**

**Основная образовательная программа 09.03.02**

**«Информационные системы и технологии»**

**Отчет по дисциплине «Структурное программирование»**

**по лабораторной работе № 2**

**по теме: «Программирование с использованием функций»**

**Студент  
группы ИДБ-21-06**

**Музафаров К.Р.**

**Преподаватель**

**Репин С.В.**

**Москва, 2022 г.**

## ЗАДАНИЕ

### Описание

Все массивы – динамические. Размерность массивов  $n$  строк и  $m$  столбцов, если в задании не указано, что дана квадратная матрица (в этом случае размерность  $n$  строк и  $n$  столбцов) или что-то иное. В задании необходимо использовать функции (не менее двух, кроме главной). Название функции должно быть с большой буквы и должно отображать то, что данная функция выполняет: например, функция `CreateArray` создает новый массив. Нельзя использовать глобальные переменные, кроме размерности массивов. Код всех программ оформляется в определенном стиле (объясняется устно на занятии).

## ЗАДАНИЕ 1(ВАРИАНТ 12)

### Описание

Если максимальный элемент квадратной матрицы находится выше главной диагонали, транспонировать матрицу, иначе определить сумму элементов строки и столбца с номерами, равными индексам максимального элемента.

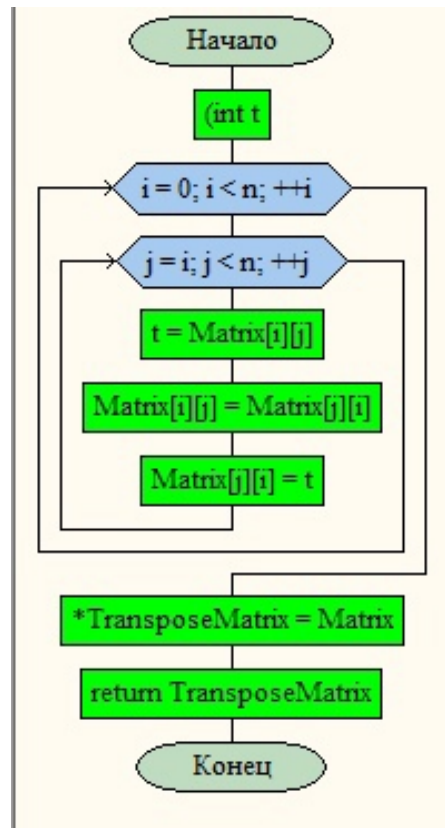
### Использованные переменные

Имя переменной	Класс	Тип	Смысл
$n$	Локальная	Целочисленный (int)	Количество строк и столбцов массива

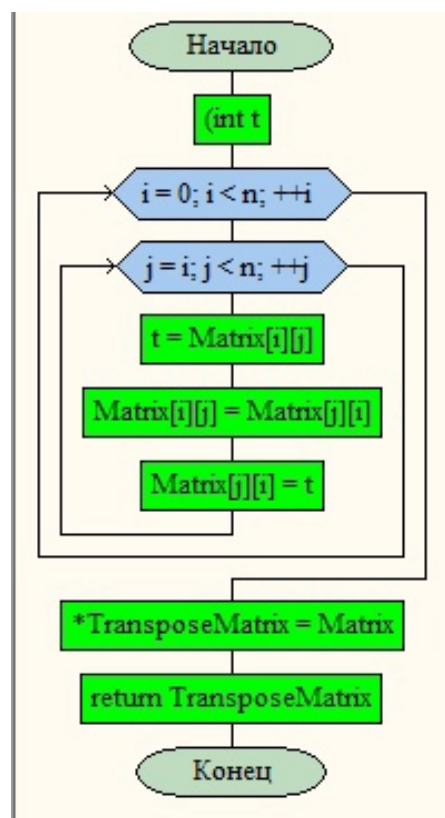
i	Локальная	Целочисленный (int)	Вспомогательная переменная для работы с массивом, счетчик интеграций цикла
j	локальная	Целочисленный (int)	Вспомогательная переменная для работы с массивом, счетчик интеграций цикла
NewMatrix	Локальная	Указатель второго порядка (int**)	Двумерный динамический массив
Matrix	Локальная	Указатель второго порядка (int**)	Двумерный динамический массив
sum	Локальная	Целочисленный (int)	Сумма элементов строки и столбца с номерами, равными индексам максимального элемента
max	Локальная	Целочисленный (int)	Максимальный элемент
indexI	Локальная	Целочисленный (int)	Строчка в которой находится максимальный элемент
indexJ	Локальная	Целочисленный (int)	Столбец в котором находится максимальный элемент

## Блок-схема

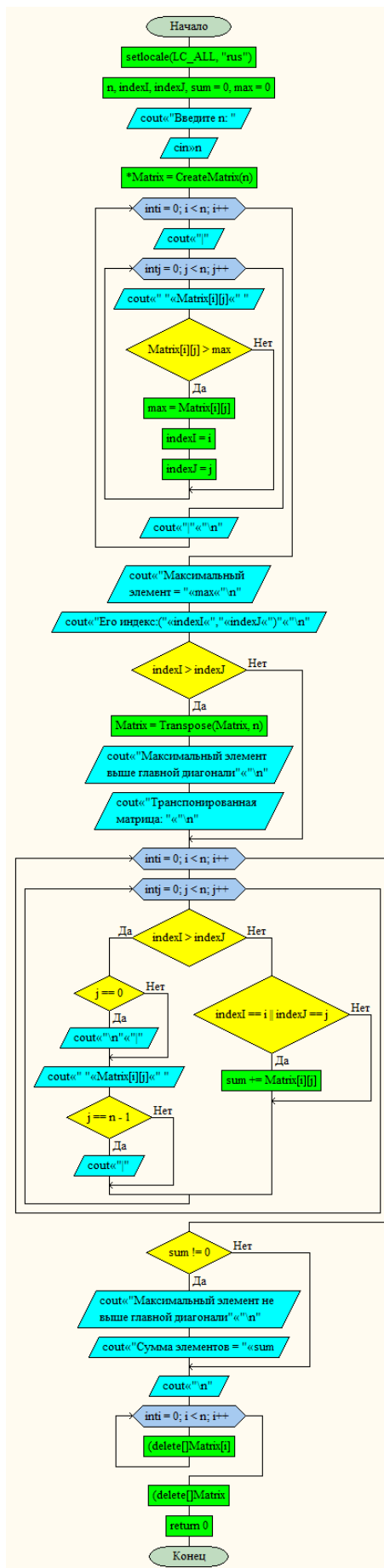
### CreateMatrix



### Transpose



# Main



## Программа

```

#include <iostream>
#include<ctime>
#include<cstdlib>
using namespace std;

int** CreateMatrix(int n)
{
    srand(time(0));
    int** NewMartix = new int* [n];
    for (int i = 0; i < n; i++) {
        NewMartix[i] = new int[n];
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            NewMartix[i][j] = (rand() % 89) + 10;
        }
    }
    return NewMartix;
}

int** Transpose(int** Matrix,int n)
{
    int t;
    for(int i = 0; i < n; ++i)
    {
        for(int j = i; j < n; ++j)
        {
            t = Matrix[i][j];
            Matrix[i][j] = Matrix[j][i];
            Matrix[j][i] = t;
        }
    }
    int** TransposeMatrix = Matrix;
    return TransposeMatrix;
}

int main()
{
    setlocale(LC_ALL, "rus");
    int n,indexI,indexJ,sum=0,max=0;
    cout << "Введите n: ";
    cin >> n;
    int** Matrix = CreateMatrix(n);
    for(int i=0; i < n; i++)
    {
        cout << "|";
        for(int j=0; j < n; j++)
        {
            cout << " " << Matrix[i][j] << " ";
            if(Matrix[i][j]>max)

```

```

        {
            max = Matrix[i][j];
            indexI = i;
            indexJ = j;
        }
    }
    cout << "|" << "\n";
}
cout << "Максимальный элемент = " << max << "\n";
cout << "Его индекс:(" << indexI << "," << indexJ << ")" << "\n" ;

if(indexI > indexJ)
{
    Matrix = Transpose(Matrix, n);
    cout << "Максимальный элемент выше главной диагонали" << "\n";
    cout << "Транспонированная матрица: " << "\n";
}
for(int i=0; i < n; i++)
{
    for(int j=0; j < n; j++)
    {
        if(indexI > indexJ)
        {
            if(j==0)
            {
                cout << "\n" << "|";
            }
            cout << " " << Matrix[i][j] << " ";
            if(j==n-1)
            {
                cout << "|";
            }
        } else
        {
            if(indexI == i || indexJ == j)
            {
                sum += Matrix[i][j];
            }
        }
    }
}
if(sum != 0)
{
    cout << "Максимальный элемент не выше главной диагонали" << "\n";
    cout << "Сумма элементов = " << sum;
}
cout << "\n";
for(int i=0; i < n; i++)
{
    delete[] Matrix[i];
}

delete[] Matrix;
return 0;
}

```

## Примеры работы программы

При  $n = 5$

```
Введите n: 5
| 37 76 90 69 77 |
| 57 61 44 98 35 |
| 86 98 69 87 60 |
| 42 58 33 84 74 |
| 29 92 95 89 16 |
Максимальный элемент = 98
Его индекс:(1,3)
Максимальный элемент не выше главной диагонали
Сумма элементов = 624
Program ended with exit code: 0
```

При  $n = 4$

```
Введите n: 4
| 79 17 22 38 |
| 31 50 82 37 |
| 28 64 16 31 |
| 53 96 10 31 |
Максимальный элемент = 96
Его индекс:(3,1)
Максимальный элемент выше главной диагонали
Транспонированная матрица:

| 79 31 28 53 |
| 17 50 64 96 |
| 22 82 16 10 |
| 38 37 31 31 |
Program ended with exit code: 0
```

При  $n = 3$

```
Введите n: 3
| 73 32 92 |
| 42 95 92 |
| 95 38 38 |
Максимальный элемент = 95
Его индекс:(1,1)
Максимальный элемент не выше главной диагонали
Сумма элементов = 299
Program ended with exit code: 0
```



## ЗАДАНИЕ 2(ВАРИАНТ 3)

### Описание

Найти максимальный среди отрицательных элементов прямоугольной матрицы и минимальный среди положительных. Если они отличаются по модулю меньше, чем на заданную величину, заменить все отрицательные элементы матрицы их модулями.

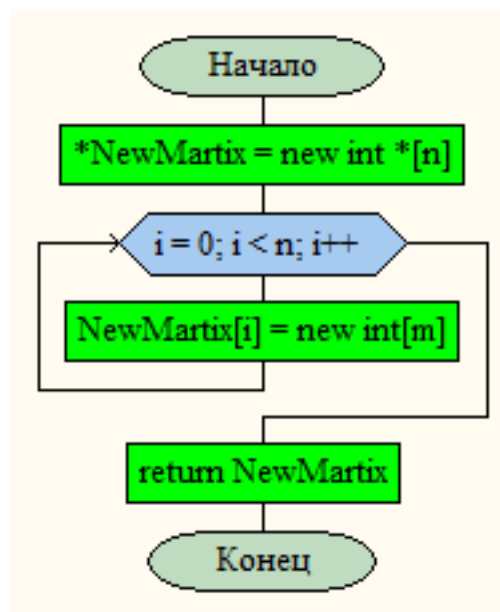
### Использованные переменные

Имя переменной	Класс	Тип	Смысл
n	Локальная	Целочисленный (int)	Количество строк массива
m	Локальная	Целочисленный (int)	Количество столбцов массива
i	Локальная	Целочисленный (int)	Вспомогательная переменная для работы с массивом, счетчик интеграций цикла
j	локальная	Целочисленный (int)	Вспомогательная переменная для работы с массивом, счетчик интеграций цикла
NewMatrix	Локальная	Указатель второго порядка (int**)	Двумерный динамический массив
Matrix	Локальная	Указатель второго порядка (int**)	Двумерный динамический массив
val	Локальная	Целочисленный (int)	Переменная для расчётов
min	Локальная	Целочисленный (int)	Минимальный положительный элемент

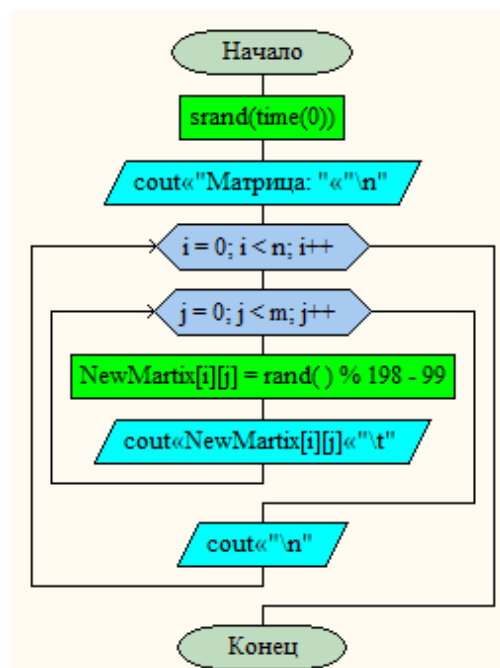
max	Локальная	Целочисленный (int)	Максимальный отрицательный элемент
diff	Локальная	Целочисленный (int)	Разница значений(по модулю)

### Блок-схема

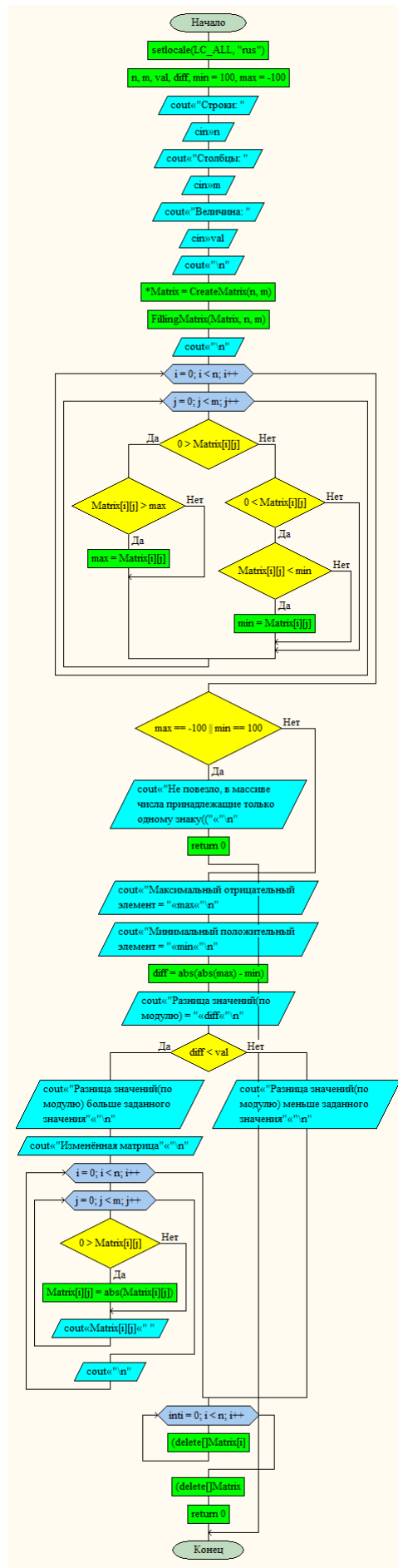
#### CreateMatrix



#### FillingMatrix



## Main



## Программа

```

#include<iostream>
#include<ctime>
#include<cstdlib>
#include<cmath>
using namespace std;
int** CreateMatrix(int n, int m)
{
    int** NewMartix = new int* [n];
    for (int i = 0; i < n; i++)
    {
        NewMartix[i] = new int[m];
    }

    return NewMartix;
}
void FillingMatrix(int** NewMartix, int n, int m)
{
    srand(time(0));
    cout << "Матрица: " << "\n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            NewMartix[i][j] = rand() % 198 - 99;
            cout << NewMartix[i][j] << "\t";
        }
        cout << "\n";
    }
}
int main()
{
    setlocale(LC_ALL, "rus");
    int n,m,val,diff,min=100,max=-100;
    cout << "Строки: ";
    cin >> n;
    cout << "Столбцы: ";
    cin >> m;
    cout << "Величина: " ;
    cin >> val;
    cout << "\n";
    int** Matrix = CreateMatrix(n, m);
    FillingMatrix(Matrix, n, m);
    cout << "\n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (0 > Matrix[i][j])
            {
                if (Matrix[i][j] > max)

```

```

        {
            max= Matrix[i][j];
        }
    } else if (0 < Matrix[i][j])
    {
        if (Matrix[i][j] < min)
        {
            min = Matrix[i][j];
        }
    }
}
}
if (max == -100 || min == 100)
{
    cout << "Не повезло, в массиве числа принадлежащие только одному знаку((" << "\n";
    return 0;
}
cout << "Максимальный отрицательный элемент = " << max << "\n";
cout << "Минимальный положительный элемент = " << min << "\n";
diff = abs(abs(max) - min);
cout << "Разница значений(по модулю) = " << diff << "\n";
if (diff < val)
{
    cout << "Разница значений(по модулю) больше заданного значения" << "\n";
    cout << "Изменённая матрица" << "\n";
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (0 > Matrix[i][j])
            {
                Matrix[i][j] = abs(Matrix[i][j]);
            }
            cout << Matrix[i][j] << " ";
        }
        cout << "\n";
    }
}

} else {
    cout << "Разница значений(по модулю) меньше заданного значения" << "\n";
}
for(int i=0; i < n; i++)
{
    delete[] Matrix[i];
}

delete[] Matrix;
return 0;
}

```

## Примеры работы программы

При  $n = 5$ ,  $m = 4$ ,  $val = 8$

```

Строки: 5
Столбцы: 4
Величина: 8

Матрица:
21  27  2   14
-42 -77 -18 -88
-94 -92 -18  43
95  -74 88  -59
8   95 -46 -80

Максимальный отрицательный элемент = -18
Минимальный положительный элемент = 2
Разница значений(по модулю) = 16
Разница значений(по модулю) меньше заданного значения
Program ended with exit code: 0

```

При  $n = 1$ ,  $m = 1$ ,  $val = 3$

```

Строки: 1
Столбцы: 1
Величина: 3

Матрица:
-56

Не повезло, в массиве числа принадлежащие только одному
знаку((
Program ended with exit code: 0

```

При  $n = 5$ ,  $m = 6$ ,  $val = 3$

```

Строки: 5
Столбцы: 6
Величина: 3

Матрица:
-69 -93 -66 -81 37 -24
-40 48 61 -14 92 -58
-70 -23 11 -71 37 -50
-62 76 44 -16 39 -85
49 38 41 -37 -11 -35

Максимальный отрицательный элемент = -11
Минимальный положительный элемент = 11
Разница значений(по модулю) = 0
Разница значений(по модулю) больше заданного значения
Изменённая матрица
69 93 66 81 37 24
40 48 61 14 92 58
70 23 11 71 37 50
62 76 44 16 39 85
49 38 41 37 11 35
Program ended with exit code: 0

```

### Вывод

В этой лабораторной работе была написана программа, которая, если максимальный элемент квадратной матрицы находится выше главной диагонали, транспонирует матрицу, иначе определяет сумму элементов строки и столбца с номерами, равными индексам максимального элемента. А так же находит максимальный среди отрицательных элементов прямоугольной матрицы и минимальный среди положительных. Если они отличаются по модулю меньше, чем на заданную величину, заменяет все отрицательные элементы матрицы их модулями. Для решения данной задачи были применены двумерные динамические массивы, а так же функции.