



## ОТЧЕТ

о выполнении лабораторной работы № 5  
по дисциплине «Языки и методы программирования»

по теме ОБРАБОТКА МНОГОМЕРНЫХ МАССИВОВ

Преподаватель	К.Т.Н., доцент	А.Г. Золин		
	(должность)	(подпись)	(дата)	(инициалы, фамилия)
Преподаватель	К.Т.Н., доцент	Е.А. Халикова		
	(должность)	(подпись)	(дата)	(инициалы, фамилия)
Студент			9.11.21	А. Е. Скрипкин
	(группа)	(подпись)	(дата)	(инициалы, фамилия)

Самара 2021 г.

## **Цель и задачи работы**

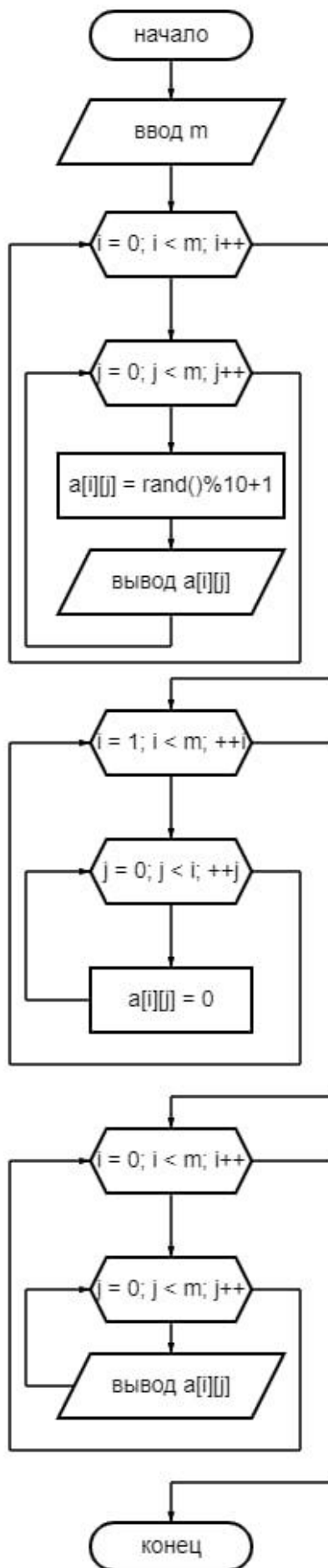
Целью работы является получение обучающимися навыков работы со средствами обработки двумерных массивов в языке C++.

Задачами работы являются:

- написание программы на языке C++ с использованием двумерных массивов;
- тестирование работоспособности программы для различных исходных данных.

**Задание 1.** Дана квадратная матрица размера  $M$ . Заменить нулями элементы матрицы, лежащие ниже главной диагонали.

## Графическая схема алгоритма:



**Программный код:**

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main() {
```

```
    cout << "enter m: ";
```

```
    int m;
```

```
    cin >> m;
```

```
    cout << endl;
```

```
    double a[m][m];
```

```
    //filling array(creating matrix)
```

```
    for(int i=0; i< m; i++) {
```

```
        for(int j=0; j<m;j++) {
```

```
            a[i][j] = rand()%10+1;
```

```
            //printing values as matrix
```

```
            cout << setw(2) << a[i][j] << " ";
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
    cout << endl;
```

```
//replacing values in matching indexes
```

```
for (int i = 1; i < m; ++i)
```

```
{
```

```
    for (int j = 0; j < i; ++j) {
```

```
        a[i][j] = 0;
```

```
    }
```

```
}
```

```
//printing matrix
```

```
for (int i=0; i < m ;i++){
```

```
    for (int j=0; j < m; j++){
```

```
        cout << setw(2) << a[i][j] << " ";
```

```
    }
```

```
    cout << endl;
```

```
}
```

```
}
```

**Примеры работы программы:**

```
enter m: 10
```

```
8 10 4 9 1 3 5 9 4 10
1 6 3 3 8 4 8 10 1 3
4 10 10 8 1 4 10 9 7 6
8 7 3 8 1 4 10 10 10 2
8 3 4 7 6 6 9 2 5 8
2 4 9 5 9 1 5 7 1 4
3 7 10 5 2 4 8 9 9 4
9 2 6 4 6 5 4 7 6 10
6 5 10 2 8 6 6 5 2 9
9 4 6 3 3 7 7 8 9 5
```

```
8 10 4 9 1 3 5 9 4 10
0 6 3 3 8 4 8 10 1 3
0 0 10 8 1 4 10 9 7 6
0 0 0 8 1 4 10 10 10 2
0 0 0 0 6 6 9 2 5 8
0 0 0 0 0 1 5 7 1 4
0 0 0 0 0 0 8 9 9 4
0 0 0 0 0 0 0 7 6 10
0 0 0 0 0 0 0 0 2 9
0 0 0 0 0 0 0 0 0 5
```

```
Program ended with exit code: 0
```

```
enter m: 5
```

```
8 10 4 9 1
3 5 9 4 10
1 6 3 3 8
4 8 10 1 3
4 10 10 8 1
```

```
8 10 4 9 1
0 5 9 4 10
0 0 3 3 8
0 0 0 1 3
0 0 0 0 1
```

```
Program ended with exit code: 0
```

### **Вывод:**

В процессе работы я получил навыки работы с двумерными массивами на языке C++.

Контрольные вопросы:

- 1) Многомерные массивы – поименованная структура данных, хранящая набор значений (элементов массива), идентифицируемых по набору индексов, принимающих целые значения из некоторого заданного непрерывного диапазона.

Определение многомерных массивов совпадает с определением одномерных массивов, за исключением того, что вместо одной размерности может быть указано несколько:

```
const int COLS = 3, ROWS = 5;  
int array[COLS][ ROWS];
```

- 2) Обращение к элементам двумерного массива осуществляется так же, как и к элементам одномерного:

ИмяМассива[ЗначениеИндекса][ЗначениеИндекса];

- 3) Индексы нужны для каждого нового измерения в массиве

4) `int a[2][2] = { {2,5},{0,1} };`

`a[0][0]` //значение равно элементу в первой строке первом столбце = 2

также часто используют вложенные циклы для заполнения или работы с данными многомерных массивов:

```
for (i = 0; i < ROWS; i++) {  
    for (j = 0; j < COLS; j++) {  
        cout << setw(7) << A[i][j];  
    }  
    cout << endl;  
}
```

- 5) вот пример организации вложенного цикла

```
for (i = 0; i < ROWS; i++) {  
    for (j = 0; j < COLS; j++) {  
        cout << setw(7) << A[i][j];  
    }  
    cout << endl;  
}
```

- 6) например так:

```
int *a = new int[ROWS * COLS];
```

- 7) Например:

```
int **a;
```

```
float **m;
```

где \*\* - двойной указатель

При формировании двумерного динамического массива сначала

выделяется память для массива указателей на одномерные массивы, а затем в цикле с параметром выделяется память под каждый одномерный массив отдельно.

Синтаксис выделения памяти под массив указателей следующий:

ИмяМассива = new Тип\* [Кол-воЭлементов]; Синтаксис выделения памяти для массива значений:

ИмяМассива[ЗначИндекса] = new Тип[Кол-воЭлементов];

ИмяМассива – идентификатор массива, то есть имя двойного указателя для выделяемого блока памяти.

Тип – тип указателя на массив.

Кол-воЭлементов – задает количество элементов (размерность) массива. Например:

```
int ROWS = 3; // количество строк
int COLS = 5; // количество элементов в строке
int i;
float** matr; //указатель для массива указателей
//выделение динамической памяти под массив указателей
matr = new float* [ROWS];
for (int i = 0; i < ROWS; i++)
    //выделение динамической памяти массива значений
    matr[i] = new float[COLS];
```