**Лабораторная работа №6**

**студента группы ИТ-211**

**Калачева Даниила Владиславовича**

**Выполнение\_\_\_\_\_\_\_\_\_\_\_\_Защита\_\_\_\_\_\_\_\_\_\_\_\_**

**ОБРАБОТКА ДИНАМИЧЕСКИХ МАССИВОВ**

**Цель работы:** ознакомиться с организацией многомерных динамических массивов в языке С/С++; приобрести практические навыки в применении односвязных линейных списков при обработке данных из внешних файлов.

**Содержание работы**

Выбрать алгоритм, составить его блок-схему и программу, выполняющую создание и обработку двумерного динамического массива, в соответствии со своим вариантом задания. Во всех вариантах предполагается, что размерность массива задается на этапе выполнения пользователем. Элементы матрицы вводятся с клавиатуры. На экран выводится исходная матрица и результаты работы программы.

**Вариант 7**: Дана матрица А(nxn), все элементы которой различны. Найти наибольший из элементов матрицы. Произвести циклический сдвиг сначала строки, а затем столбца, в которых он находится так, чтобы переместить элемент в центр матрицы.

**Ход работы**

**Часть А**

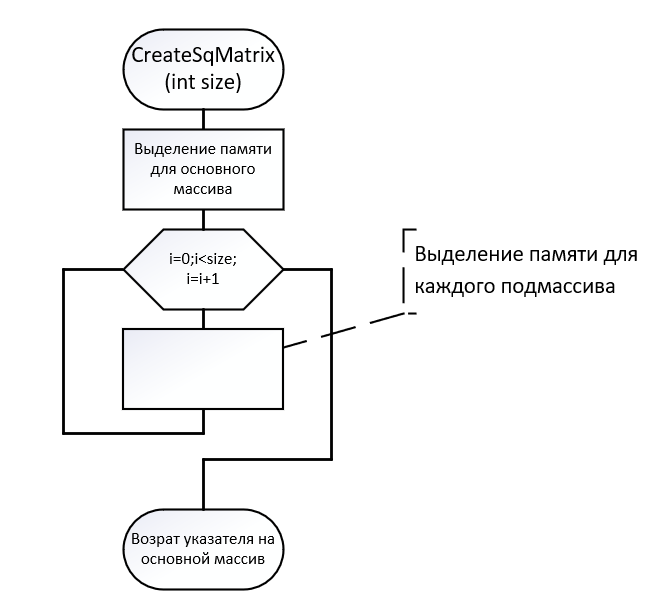
1. Для решения поставленной задачи, составил блок-схемы функций и их спецификации:

**Функция CreateSqMatrix** (см. Рисунок 1)

**Назначение:** создать динамический двумерный массив, предварительно выделив под него память

**Входные данные**: количество строк матрицы

**Выходные данные**: указатель на двумерный массив



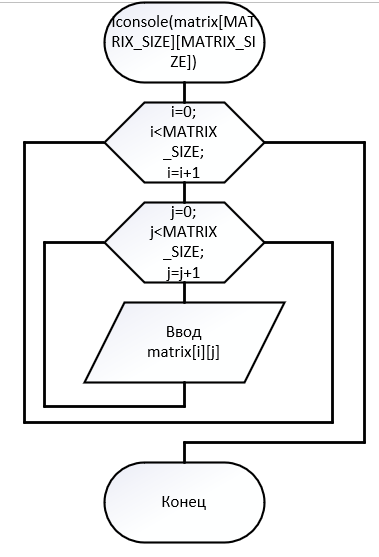
1. Функция CreateSqMatrix

**Функция iconsole** (см. Рисунок 2)

**Назначение:** запись в двумерный динамический массив через консоль

**Входные данные**: указатель на двумерный динамический массив, его размер

**Выходные данные**: элементы матрицы



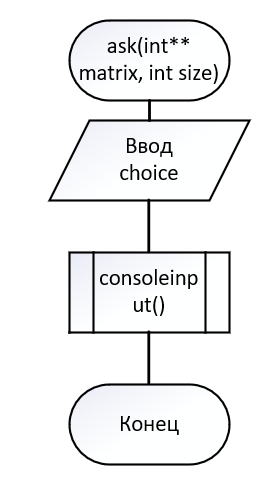
1. Функция iconsole

**Функция ask** (см. Рисунок 3)

**Назначение:** выбор ввода и вызов функций ввода

**Входные данные**: двумерный массив фиксированного размера

**Выходные данные**: отсутствуют



1. Функция a ask

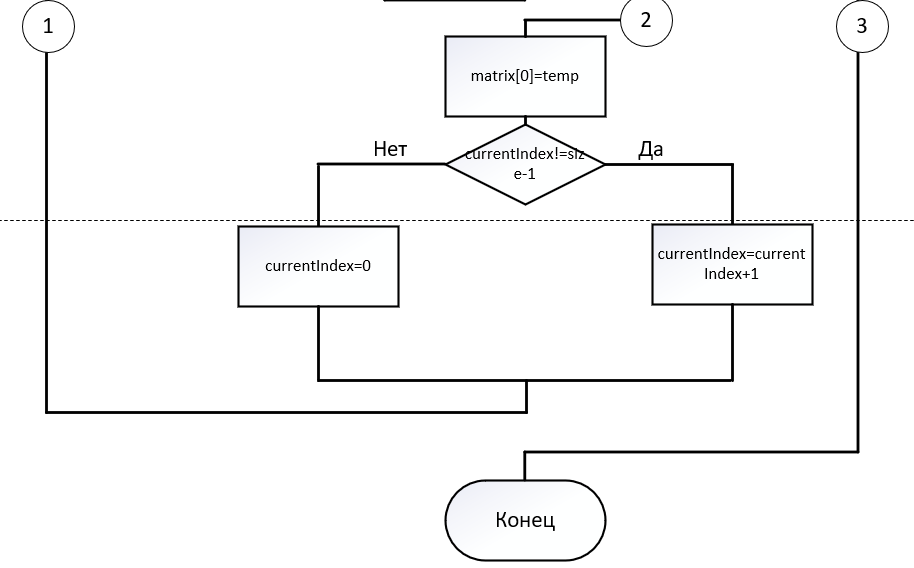
**Функция MoveRows** (см. Рисунок 4)

**Назначение:** циклический сдвиг строк

**Входные данные**: указатель на двумерный диамический массив, размер одного измерения переданного массива, индекс сдвигаемой строки, индекс строки до которой происходит сдвиг

**Выходные данные**: отсутствуют





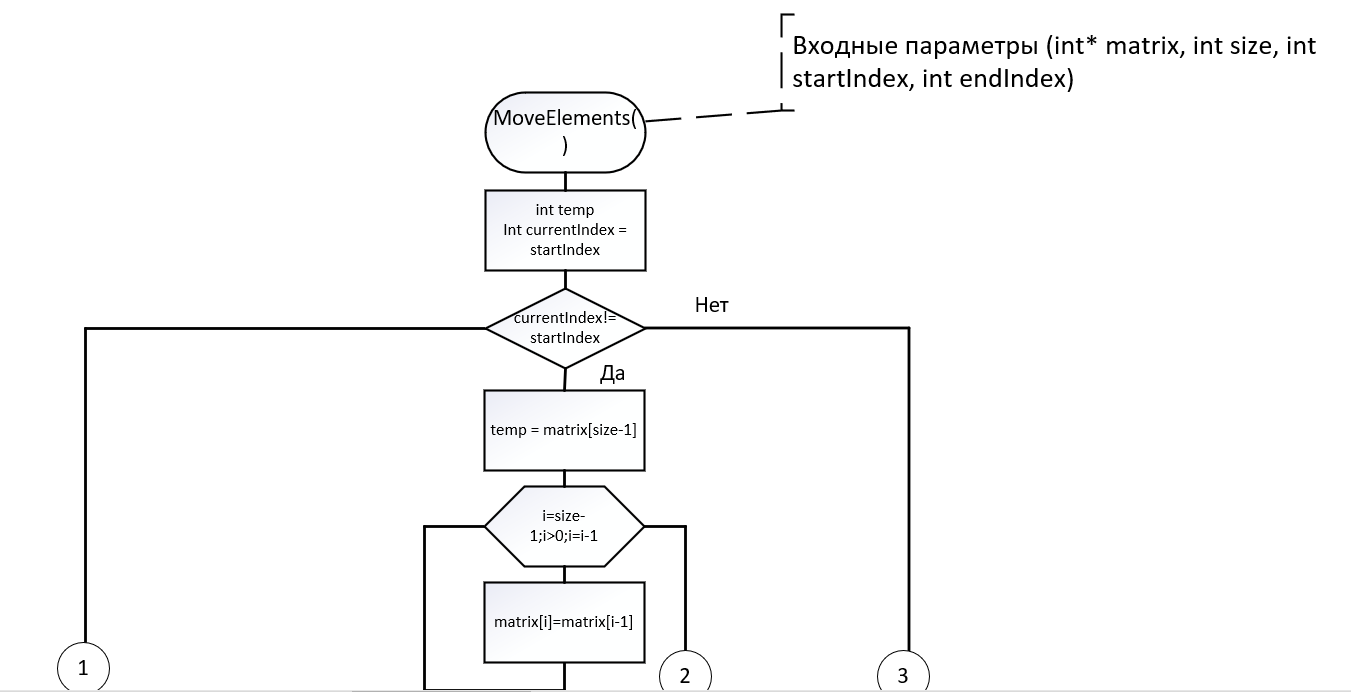
1. Функция MoveRows

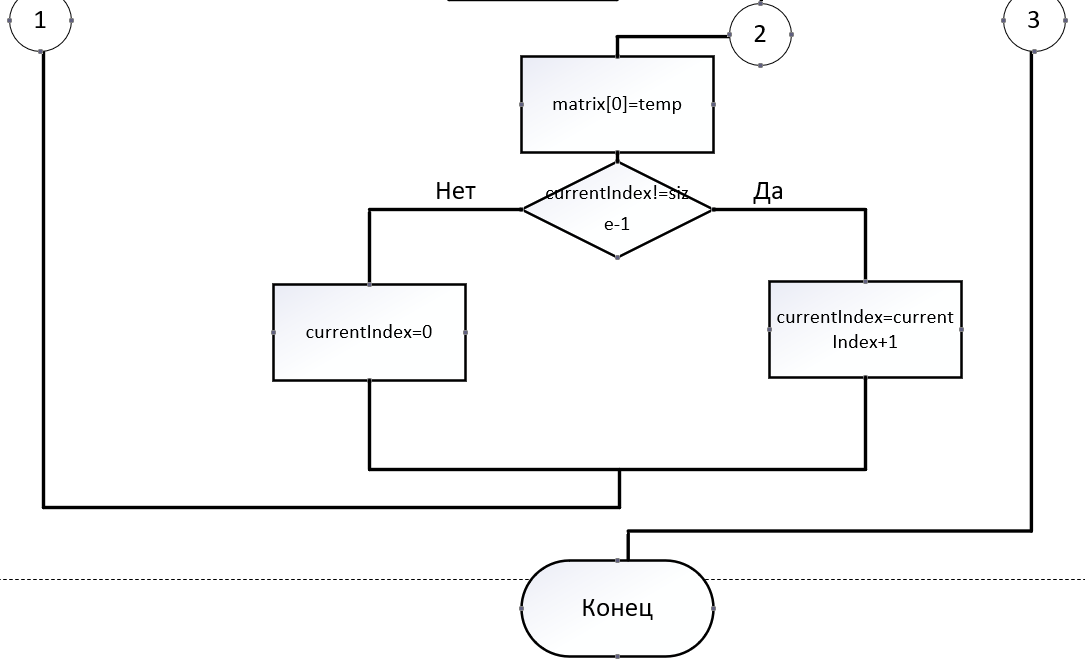
**Функция MoveElements** (см. Рисунок 5)

**Назначение:** циклический сдвиг элементов массива

**Входные данные**: указатель на массив, размер переданного массива, индекс сдвигаемого элемента, индекс столбца до которого происходит сдвиг

**Выходные данные**: отсутствуют





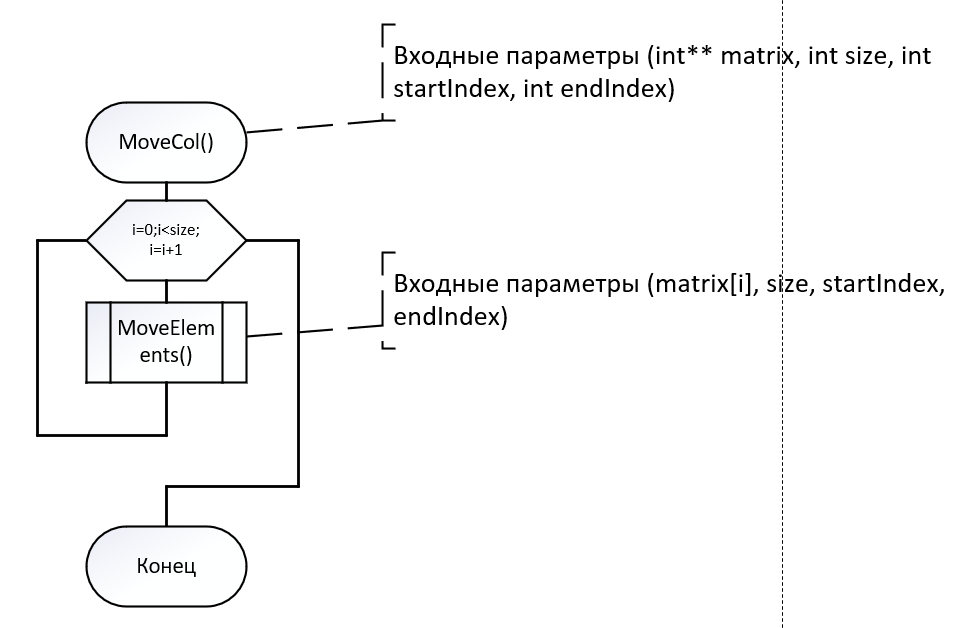
1. Функция MoveElements

**Функция MoveCol** (см. Рисунок 6)

**Назначение:** циклический сдвиг столбцов

**Входные данные**: указатель на двумерный динамический массив, размер одного измерения переданного массива, индекс сдвигаемой строки, индекс строки до которой происходит сдвиг

**Выходные данные**: отсутствуют



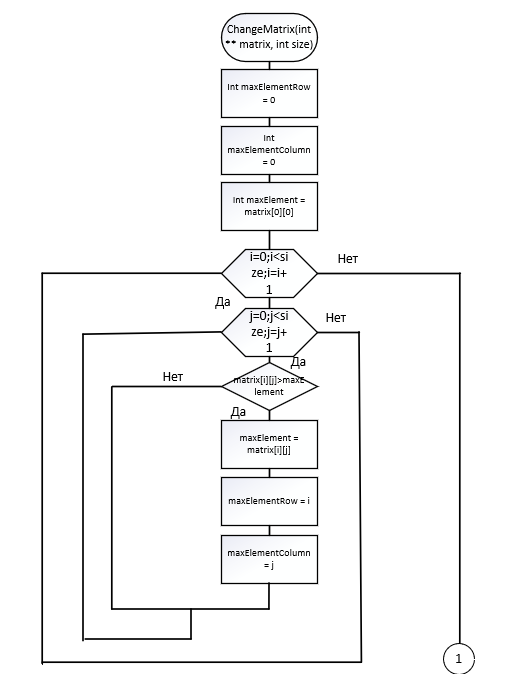
1. Функция MoveCol

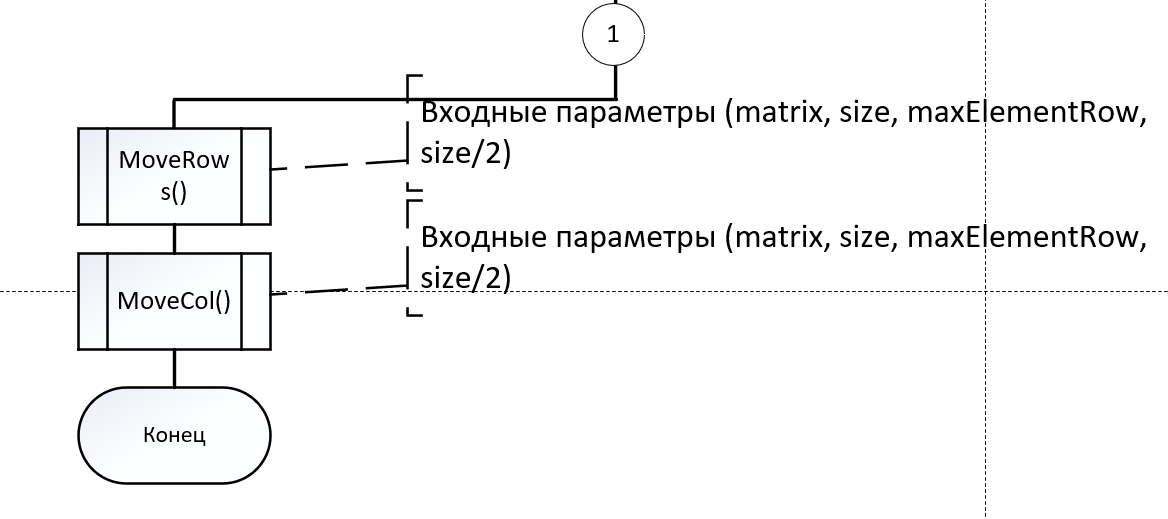
**Функция ChangeMatrix** (см.Рисунок 7)

**Назначение:** поместить максимальный элемент в центр матрицы с помощью циклических сдвигов по строкам, затем по столбцам

**Входные данные**: указатель на динамический двумерный массив, его размер

**Выходные данные**: отсутствуют





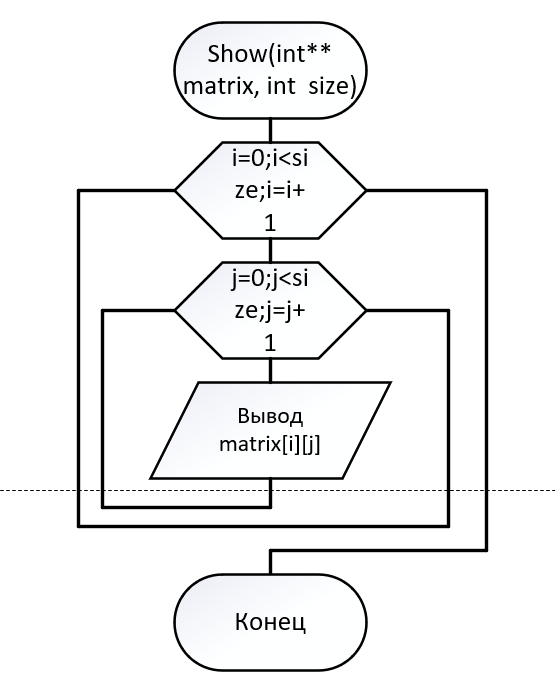
1. Функция ChangeMatrix

**Функция Show** (см.Рисунок 8)

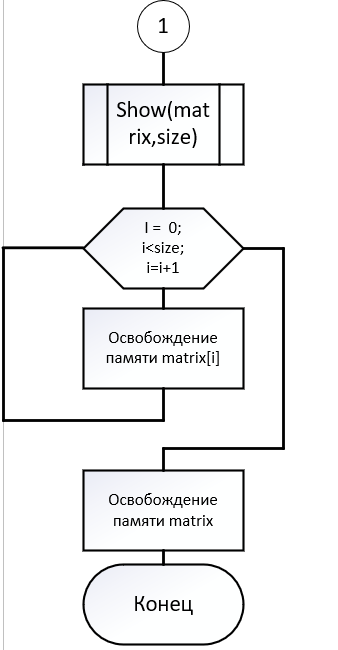
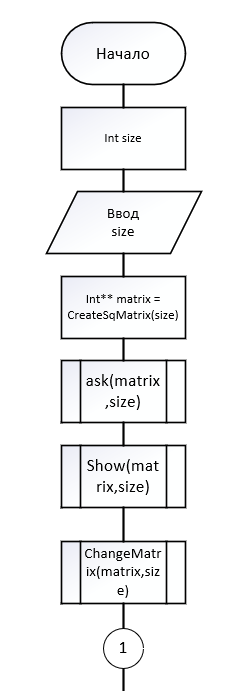
**Назначение:** вывод двумерного динамического массива

**Входные данные**: указатель на двумерный динамический массив, его размер

**Выходные данные**: отсутствуют



1. Функция Show



1. Основная программа
2. Написал код программы на языке С++.

#include <iostream>

#include <locale>

#include <fstream>

using namespace std;

int\*\* CreateSqMatrix(int size)

{

int\*\* matrix;

matrix = new int\* [size];

for (int i = 0; i < size; i++)

matrix[i] = new int[size];

return matrix;

}

void iconsole(int\*\* matrix, int size)

{

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

cout << "Введите " << i << " элемент " << j << " строки матрицы\n";

cin >> matrix[i][j];

}

}

}

void ask(int\*\* matrix, int size)

{

int choice;

cout << "Введите 1 для ввода через консоль.\n";

cin >> choice;

if (choice == 1)

iconsole(matrix, size);

}

void MoveRows(int\*\* matrix, int size, int startIndex, int endIndex)

{

int currentIndex = startIndex;

while (currentIndex != endIndex)

{

int\* temp = new int[size];

temp = matrix[size - 1];

for (int i = size - 1; i > 0; i--)

{

matrix[i] = matrix[i - 1];

}

matrix[0] = temp;

if (currentIndex != size - 1)

currentIndex++;

else

currentIndex = 0;

}

}

void MoveElements(int\* matrix, int size, int startIndex, int endIndex)

{

int temp;

int currentIndex = startIndex;

while (currentIndex != endIndex)

{

temp = matrix[size - 1];

for (int i = size - 1; i > 0; i--)

{

matrix[i] = matrix[i - 1];

}

matrix[0] = temp;

if (currentIndex != size - 1)

currentIndex++;

else

currentIndex = 0;

}

}

void MoveCol(int\*\* matrix, int size, int startIndex, int endIndex)

{

for (int i = 0; i < size; i++)

MoveElements(matrix[i], size, startIndex, endIndex);

}

void ChangeMatrix(int\*\* matrix, int size)

{

int maxElementRow = 0;

int maxElementColumn = 0;

int maxElement = matrix[0][0];

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

if (matrix[i][j] > maxElement)

{

maxElement = matrix[i][j];

maxElementRow = i;

maxElementColumn = j;

}

}

}

MoveRows(matrix, size, maxElementRow, size / 2);

MoveCol(matrix, size, maxElementColumn, size / 2);

}

void Show(int\*\* matrix, int size)

{

cout << endl << endl;

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

cout << matrix[i][j] << "\t";

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "ru");

int size;

cout << "Введите размер одного измерения квадратной матрицы: ";

cin >> size;

int\*\* matrix = CreateSqMatrix(size);

ask(matrix, size);

Show(matrix, size);

ChangeMatrix(matrix, size);

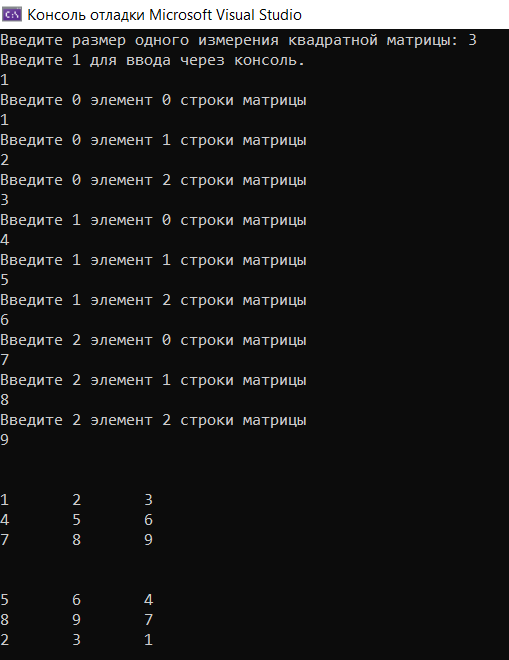
Show(matrix, size);

for (int i = 0; i < size; i++)

delete[] matrix[i];

delete[] matrix;  
}

1. Результаты работы программы. (см. Рисунок 6)



1. Результаты работы программы

**Вывод:** в среде разработки Visual Studio на языках С и С++ осуществлена удобная работа с динамическими массивами. Работа с ними значительно лучше, чем со статическими, так как после обработки можно очищать память нужной командой и адаптировать массив под различные задачи.