**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**Отчет По Дисциплине**

**«Алгоритмизация и программирование»**

**Лабораторная работа №5**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-113

Таранов Евгений Николаевич

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ФИО)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ФИО)

Москва 2021 г.

**Задание №6**

Задание: Задана матрица целых чисел. Определить максимальный элемент главной диагонали и записать нули в строку в которой он находится.

1. Таблица имён:

|  |  |  |
| --- | --- | --- |
| Исходные данные | | |
| a[x][x] | Целочисленный | Двумерный массив |
| size | Целочисленный | Размерность массива |
| Рабочие переменные | | |
| maximum | Целочисленный | Максимальный елемент |
| row | Целочисленный | Номер рядка с максимальным елементом |
| Результат | | |
| a[x][x] | Целочисленный | Двумерный массив с нулями в рядке где был максимальный елемент |

1. Блок-схема:



Рис. 1 – Алгоритм программы

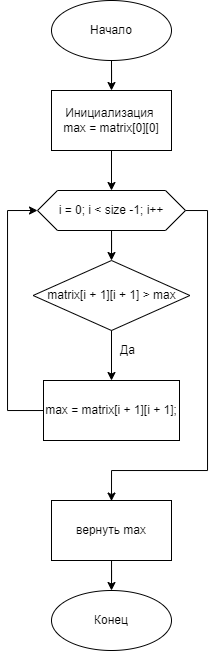


Рис. 2 – Алгоритм поиска максимального елемента главной диагнали

1. Отладочный пример:

.

.

.

1. Код программы:

#include <iostream>

using namespace std;

//заполнение матрицы

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

{

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

{

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

{

matrix[i][j] = rand() % 15;

}

}

}

//печать матрицы

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

{

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

{

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

{

cout << matrix[i][j] <<'\t';

}

cout << endl << endl;

}

}

//максимальный елемент в главной диагонали

int max(int\*\* matrix, int size, int &row)

{

int max = matrix[0][0];

int maxRow = 0;

for (int i = 0; i < size - 1; i++)

{

if (matrix[i + 1][i + 1] > max)

{

max = matrix[i + 1][i + 1];

maxRow = i + 1;

}

}

row = maxRow;

return max;

}

//заполнения рядка с макисмальным елементом

void fillRowZero(int\*\* matrix, int size, int& row)

{

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

{

matrix[row][i] = 0;

}

}

int main()

{

setlocale(LC\_ALL, "ru");

srand(time(NULL));

int size;

cout << "Введите размерность квадратной матрицы: ";

cin >> size;

int\*\* matrix;

matrix = new int\* [size];

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

{

matrix[i] = new int[size];

}

int maximum, row;

fillMatrix(matrix, size);

cout << "Матрица до обработки: " << endl << endl;

printMatrix(matrix, size);

cout << endl;

maximum = max(matrix, size, row);

cout << "Максимальный елемент = " << maximum << endl;

cout << "Рядок: " << row + 1 << endl << endl;

fillRowZero(matrix, size, row);

cout << "Матрица после обработки: " << endl << endl;

printMatrix(matrix, size);

}

1. Результат выполнения работы программы:

Для тестирования программы была создана матрица 6 на 6. Результат работы можно увидеть на скриншоте.

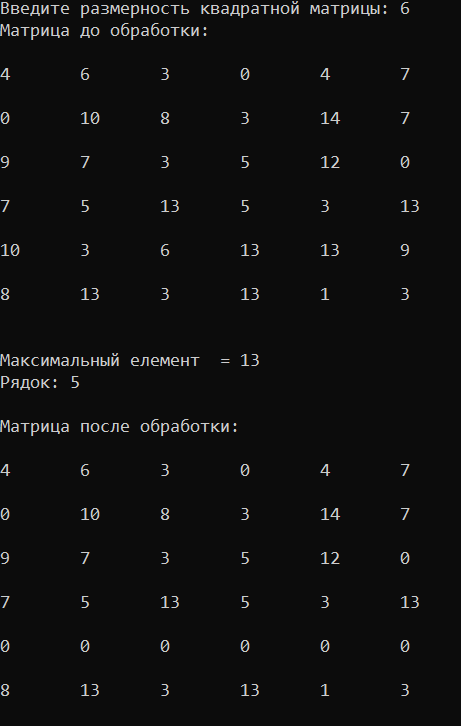


Рис. 3 – Результат работы

1. Вывод:

В ходе выполнения работы работал с двумерным динамическим массивом. Программа находила максимальный элемент в главной диагонали матрицы и заполняла рядок нулями, в котором он находится.