ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ПРАКТИЧЕСКОЙ РАБОТЕ №4

По дисциплине «Процедурное программирование»

Выполнил: ст. гр. ТКИ – 112

Зинченко Б.А.

Проверил: к.т.н., доц.

Васильева М.А.

Москва 2021

**Оглавление**

[ЗАДАНИЕ 4.3 3](#_Toc72361031)

ЗАДАНИЕ 4.3

1. Условие задачи

Создать многомерный массив nˣm из *n* целых чисел. Реализовать возможность заполнения массива, как случайными числами, так и с помощью клавиатуры по желанию пользователя. Пункты задания организовать в виде функций (методов). Вывести массив на экран.

1. Заменить максимальный элемент каждой строки нулем.  
2. Вставить перед всеми строками, первый элемент которых  
делится на 3, строку из нулей.

1. Блок-схема алгоритма

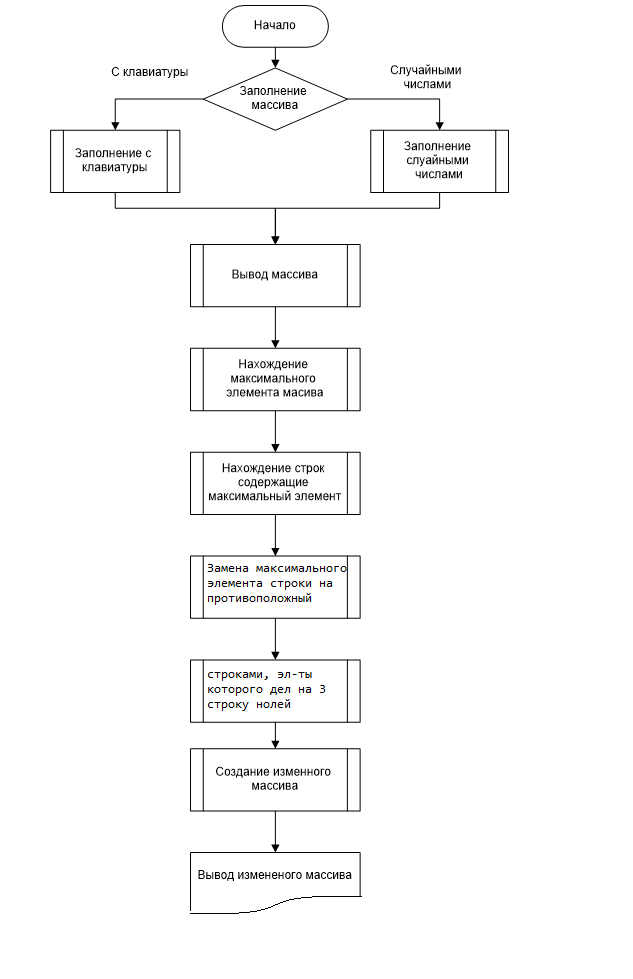


Рисунок 1 – Блок-схема 4.3

1. Текст программы на языке C++

# #include <iostream>

#include <random>

#include <sstream>

#include <iomanip>

using namespace std;

/\* \brief Функция создания массива

\* \param row количество строк

\* \param col количество стобцов

\* \return пустой массив

\*/

int\*\* initializeArray(const size\_t row, const size\_t col);

/\* \brief Функция замены элементов массива на 0

\* \param row количество строк

\* \param col количество стобцов

\* \return преобразованный массив массив

\*/

int\*\* swapToZero(int\*\* array, size\_t row, size\_t col);

/\* \brief Функция заполнения массива рандомными числами

\* \param row количество строк

\* \param col количество стобцов

\* \return Заполненный массив

\*/

void randomizeArray(int\*\* matrix, const size\_t row, const size\_t col, const int min, const int max);

/\* \brief Функция заполнения массива мануальными числами

\* \param row количество строк

\* \param col количество стобцов

\* \return Заполненный массив

\*/

void manualInput(int\*\* matrix, const size\_t row, const size\_t col);

/\* \brief Функция преобразования массива в строчку

\* \param row количество строк

\* \param col количество стобцов

\* \return Заполненный массив

\*/

string arrayToString(int\*\* matrix, const size\_t row, const size\_t col, const int width, const string& message = "");

/\* \brief Функция проверки деления элементов массива на 3

\* \param num элемент массива

\* \return делится ли этот элемент на 3

\*/

bool divisible(const int num);

/\* \brief Функция получения нового количества столбцов

\* \param row количество строк

\* \param col количество стобцов

\* \return новое количество столбцов

\*/

size\_t getNewColSize(int\*\* array, const size\_t row, const size\_t col);

/\* \brief Функция создания нового массива

\* \param row количество строк

\* \param старый массив

\* \param col количество стобцов

\* \return новый массив массив

\*/

int\*\* insertColumns(int\*\* array, const size\_t row, const size\_t col, const size\_t newCol);

/\*\*

\* \brief Точка входа в программу

\* \return Возвращает нуль, в случае успеха

\*/

int main()

{

const int MIN = -99, MAX = 99, WIDTH = 3;

size\_t row, col;

cout << "Please input matrix dimensons\nrow x col: ";

cin >> row >> col;

int\*\* array = initializeArray(row, col);

unsigned int mode;

cout << "if 1 then random, if 2 then manual";

cin >> mode;

if (mode == 1)

{

randomizeArray(array, row, col, MIN, MAX);

}

else if (mode == 2)

{

manualInput(array, row, col);

}

else {

cout << "Try again ...";

}

swapToZero(array, row, col);

cout << arrayToString(array, row, col, WIDTH) << "\n\n";

size\_t col2 = getNewColSize(array, row, col);

int\*\* array2 = insertColumns(array, row, col, col2);

cout << arrayToString(array2, row, col2, WIDTH) << "\n\n";

return 0;

}

int\*\* initializeArray(const size\_t row, const size\_t col)

{

int\*\* matrix = new int\* [row];

for (size\_t i = 0; i < row; i++)

{

matrix[i] = new int[col];

}

return matrix;

}

int\*\* swapToZero(int\*\* array, size\_t row, size\_t col) {

int maxElement = array[0][0];

int maxI = 0;

int maxJ = 0;

for (size\_t i = 0; i < row; i++)

{

for (size\_t j = 0; j < col; j++)

{

if (array[i][j] >= maxElement)

{

maxElement = array[i][j];

maxI = i;

maxJ = j;

}

}

array[maxI][maxJ] = 0;

}

return array;

}

void randomizeArray(int\*\* matrix, const size\_t row, const size\_t col, const int min, const int max)

{

std::random\_device rd;

std::mt19937 gen(rd());

const uniform\_int\_distribution<> uniformIntDistribution(min, max);

for (size\_t i = 0; i < row; ++i)

for (size\_t j = 0; j < col; ++j)

matrix[i][j] = rand() % 10;

}

void manualInput(int\*\* matrix, const size\_t row, const size\_t col)

{

for (size\_t i = 0; i < row; i++)

for (size\_t j = 0; j < col; j++)

{

cout << "array[" << i << ", " << j << "] = ";

cin >> matrix[i][j];

}

}

string arrayToString(int\*\* matrix, const size\_t row, const size\_t col, const int width, const string& message) //{{1,2},{3,4},{5,6}}

{

std::stringstream buffer;

buffer << message;

for (size\_t i = 0; i < row; ++i)

{

if (i != row && i != 0)

buffer << "\n";

buffer << "{";

for (size\_t j = 0; j < col; ++j)

{

buffer << setw(width) << matrix[i][j];

if (j != col - 1)

buffer << ", ";

}

buffer << "}";

}

return buffer.str();

}

bool divisible(const int num)

{

return num % 3 == 0;

}

size\_t getNewColSize(int\*\* array, const size\_t row, const size\_t col)

{

size\_t newCol = col;

for (size\_t j = 0; j < col; j++)

if (divisible(array[0][j]))

newCol++;

return newCol;

}

int\*\* insertColumns(int\*\* array, const size\_t row, const size\_t col, const size\_t newCol)

{

int\*\* newArray = initializeArray(row, newCol);

size\_t colOffset = 0;

for (size\_t j = 0; j < col; j++)

if (divisible(array[0][j]))

{

for (size\_t i = 0; i < row; i++)

newArray[i][j + colOffset] = 0;

colOffset++;

for (size\_t i = 0; i < row; i++)

newArray[i][j + colOffset] = array[i][j];

}

else

for (size\_t i = 0; i < row; i++)

newArray[i][j + colOffset] = array[i][j];

return newArray;

}

Результаты работы программы

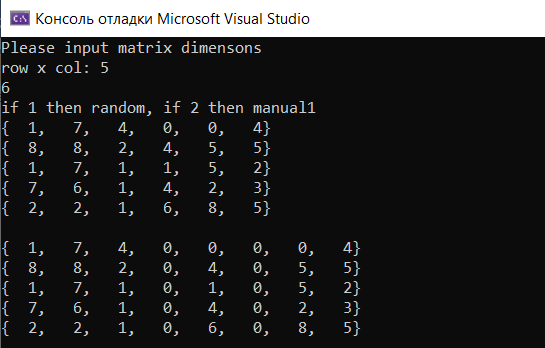


Рисунок 2 ˗ Вывод программы 4.3