Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа**

**«Динамические массивы»**

Выполнила:

студентка группы ИВТ-23-2Б

Сычева Евгения Андреевна

Проверила:

Доцент кафедры ИТАС

О. А. Полякова

2024 г.

Постановка задачи:

Написать программу, в которой создаются динамические массивы, и выполнить их обработку.

Сформировать двумерный массив. Удалить из него строку и столбец, на пересечении которых находится максимальный элемент.

Анализ задачи:

1. Создадим двумерный динамический массив, заполним его случайными числами.
2. Среди всех элементов найдем максимальный с помощью двух вложенных циклов.
3. Удалим строку, в которой находится максимальный элемент, рассматривая ее как одномерный динамический массив.
4. Используя цикл for удалим столбец, в котором находится максимальный элемент.
5. Выведем полученный массив.
6. Освободим память от «мусора», образовавшемся от преобразований над динамическим массивом.

Код программы с комментариями:

#include <iostream>

#include <cstdlib>

using namespace std;

int main() {

setlocale(LC\_ALL, "RU");

// Создаем двумерный динамический массив

int rows, cols;

cout << "Введите количество строк и столбцов: ";

cin >> rows >> cols;

int\*\* arr = new int\* [rows];

for (int i = 0; i < rows; i++) {

arr[i] = new int[cols];

}

// Инициализируем массив случайными числами и находим максимальный элемент

int maxElem = -1;

int maxRow, maxCol;

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

arr[i][j] = rand() % 100 - 50;

cout << arr[i][j] << " ";

if (arr[i][j] > maxElem) {

maxElem = arr[i][j];

maxRow = i;

maxCol = j;

}

}

cout << endl;

}

// Удаляем строку

delete[] arr[maxRow];

// Удаляем столбец

for (int i = 0; i < rows; i++) {

for (int j = maxCol; j < cols - 1; j++) {

arr[i][j] = arr[i][j + 1];

}

}

// Выводим полученный массив

cout << "Массив с удаленными элементами:" << endl;

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

for (int j = 0; j < cols - 1; j++) {

cout << arr[i][j] << " ";

}

cout << endl;

}

// Освобождаем память

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

delete[] arr[i];

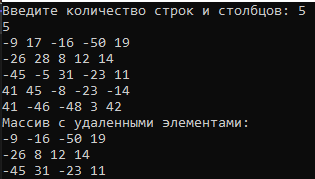
}

delete[] arr;

return 0;

}

Результат работы программы:



Блок-схема алгоритма:

Вывод:

Программа работает исправно, используя минимальное количество памяти.