Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №10**

Дисциплина: «Информатика»

Тема: Динамические массивы

Вариант 15

Выполнила:

Студент группы ИВТ-20-2б

Чувашев Максим Алексеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Организация динамических массивов.

**Постановка задачи**

Сформировать двумерный массив. Удалить из него

строку и столбец, на пересечении которых находится

максимальный элемент.

**Анализ задачи**

1. Типы данных, которые будут использованы:
   1. Тип integer:

* int rows; // Данная переменная будет хранить количество строчек в двумерном массиве.
* int cols; // Данная переменная будет хранить количество столбцов в двумерном массиве.
* int count; // Данная переменная будет отвечать за подсчет количество одинаковых максимальных элементов двумерного массива.
* int Max; // Данная переменная будет хранить значение максимального элемента максимального элемента двумерного динамического массива.
* int countRM; // Данная переменная будет отвечать за подсчет количества строк, в которых находится максимальные элементы.
* int countCM; // Данная переменная будет отвечать за подсчет количества столбцов, в которых находится максимальные элементы.
* int\*\* arr; // Массив указателей, которые указывают на динамические массивы. Это изначальный массив, размер которого задает пользователь.
* int i; // Переменная, которая отвечает за передвижение по циклу.
* int rows2; // Переменная, которая хранит количество строк нового динамического массива с учетом вычета тех строк, в которых находятся максимальные элементы массива.
* int cols2; // Переменная, которая хранит количество столбцов нового динамического массива с учетом вычета тех столбцов, в которых находятся максимальные элементы массива.
* int\*\* arr2; // Массив указателей на динамические массивы. Данный массив будет конечным массивом, который нам необходимо будет получить.
* int countR; // Поле, которое будет хранить номер строки, в которой находится максимальный элемент массива.
* int countC; // Поле, которое будет хранить номер столбца, в которой находится максимальный элемент массива.
* int j; // Переменная, отвечающая за передвижение по вложенному циклу.
* int c; // Переменная, используемая для передвижения по элементам массива arr3.
* int\* n1; // Массив, в который будут записаны все номера строк максимальных элементов массива arr.
* int\* n2; // Массив, в который будут записаны все номера столбцов максимальных элементов массива arr.
* int q; // Переменная, используемая для передвижения по строкам нового массива arr2.
* int w; // Переменная, используемая для передвижения по столбцам нового массива arr2.
* int k; // Переменная, используемая для передвижения по вложенному циклу for.
* int main(); // Главная функция, которая отвечает за объявления большинства переменных, создание динамических двумерных массивов и вызов всех остальных функций.
  1. Тип bool:
* bool f; // Переменная, используемая как условие копирования данных из одного массива в другой.
  1. Тип void:
* void countRC(int\* rows, int\* cols); // Функция, которая просит пользователя ввести количество строк и столбцов.
* void FillArr(int\*\* arr, int\* rows, int\* cols); // Функция, которая заполнят массив arr, случайными числами от 0 до 20. Так же в качестве параметра эта функция принимает количество строк и столбцов этого массива.
* void ShowArr(int\*\* arr, int\* rows, int\* cols); // Функция, которая выводит массив полученный в качестве параметра на консоль. Так же в качестве параметра эта функция принимает количество строк и столбцов этого массива.
* void countMax(int\*\* arr, int\* rows, int\* cols, int\* count, int\* Max); // Функция, которая считает количество максимальных элементов и записывает сам максимальный элемент.
* void Search(index\* arr3, int Max, int\*\* arr, int rows, int cols, int\* countRM, int\* countCM); // Функция, которая проходится по массиву arr и записывает индексы элементов равных максимальному MAX, в поля массива arr3. Далее функция создает два динамических массива n1 и n2. После чего записывает в них все индексы из массива arr3. Далее эти массивы сортируются по возрастанию и уже с отсортированными массивами ведется работа. Функция проходится по всем элементам этих массивов. Сравнивает эти элемента с предыдущим и в случае если эти элементы не совпали, то увеличивает переменную (\*countCM)++; (\*countRM)++; Так удается избежать неправильного подсчета количества строк и столбцов которые необходимо будет удалить.
* void Del(int\*\* arr, int\* rows, int\* cols, int\*\* arr2, int\* rows2, int\* cols2, int\* count, index\* arr3); // Функция отвечающая за заполнение массива arr2, с помощью массива arr, в массив arr2 записываются все элементы который не находятся в строках или столбцах в которых находится максимальный элемент MAX.
  1. Объекты структуры index:
* index\* arr3; // Динамический массив структуры index, хранит номер строки и столбца максимального элемента.

1. Какие с этими данными надо выполнить действия:
   1. void countRC(int\* rows, int\* cols);

* Данная функция принимает в качестве параметров по ссылке значения переменных rows, cols.
* Данная функция просит пользователя присвоить переменной rows значение, которое хочет пользователь. Важно чтобы данная переменная имела натуральное значение. Чтобы пользователь ввел корректное значение, функция будет запрашивать ввод через цикл while.

while (\*rows <= 0)

{

cout << "Введите количество строк: ";

cin >> \*rows;

}

* Далее функция просит пользователя присвоить переменной cols значение, которое хочет пользователь. Важно чтобы данная переменная имела натуральное значение. Чтобы пользователь ввел корректное значение, функция будет запрашивать ввод через цикл while.

while (\*cols <= 0)

{

cout << "Введите количество столбцов: ";

cin >> \*cols;

}

* Полный код функции void countRC(int\* rows, int\* cols);

void countRC(int\* rows, int\* cols)

{

while (\*rows <= 0)

{

cout << "Введите количество строк: ";

cin >> \*rows;

}

while (\*cols <= 0)

{

cout << "Введите количество столбцов: ";

cin >> \*cols;

}

}

* 1. void FillArr(int\*\* arr, int\* rows, int\* cols)
* Данная функция принимает в качестве параметров массив указателей arr, переменную rows, которая содержит количество строк двумерного динамического массива. А так же переменную cols, которая хранит количество столбцов двумерного динамического массива arr.
* Далее функция создает два цикла for для того чтобы пройтись по двумерному динамическому массиву. И заполняет все ячейки двумерного динамического массива с помощью функции rand(). Функция заполняет все ячейки фукции случайными значениями от 0 до 19.

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

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

arr[i][j] = rand() % 20;

}

}

* Полный код функции void FillArr(int\*\* arr, int\* rows, int\* cols);

void FillArr(int\*\* arr, int\* rows, int\* cols)

{

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

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

arr[i][j] = rand() % 20;

}

}

}

* 1. void ShowArr(int\*\* arr, int\* rows, int\* cols)
* Данная функция принимает в качестве параметров массив указателей на динамические массивы arr. Переменную rows и cols.
* Далее функция с помощью двух циклов for проходится по всем элементам динамического массива arr. И выводит каждый элемент данного массива на консоль с помощью оператора вывода cout.

for (int i = 0; i < \*rows; i++)

{

for (int j = 0; j < \*cols; j++)

{

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

}

cout << endl;

}

* Полный код функции void ShowArr(int\*\* arr, int\* rows, int\* cols);

void ShowArr(int\*\* arr, int\* rows, int\* cols)

{

cout << endl;

for (int i = 0; i < \*rows; i++)

{

for (int j = 0; j < \*cols; j++)

{

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

}

cout << endl;

}

}

* 1. void countMax(int\*\* arr, int\* rows, int\* cols, int\* count, int\* Max);
* Данная функция принимает в качестве параметров массив указателей на динамические массивы arr. Переменную rows и cols. А также переменную count которая будет отвечать за подсчет количества максимальных элементов. И переменную Max. В которой будет храниться значение максимального элемента массива.
* Данная функция с помощью двух вложенных циклов for проходится по всем элемента массива и ищем максимальный элемент массива и записываем это значение в переменную Max. Так же если встречается элемент равный Max, то мы увеличиваем счетчик count, если мы находим элемент который больше Max, то мы обнуляем наш счетчик.

for (int i = 0; i < \*rows; i++)

{

for (int j = 0; j < \*cols; j++)

{

if (\*Max <= arr[i][j])

{

if (\*Max == arr[i][j])

{

(\*count)++;

}

else

{

\*Max = arr[i][j];

\*count = 1;

}

}

}

}

* Полный код функции void countMax(int\*\* arr, int\* rows, int\* cols, int\* count, int\* Max);

void countMax(int\*\* arr, int\* rows, int\* cols, int\* count, int\* Max)

{

for (int i = 0; i < \*rows; i++)

{

for (int j = 0; j < \*cols; j++)

{

if (\*Max <= arr[i][j])

{

if (\*Max == arr[i][j])

{

(\*count)++;

}

else

{

\*Max = arr[i][j];

\*count = 1;

}

}

}

}

}

* 1. void Search(index\* arr3, int Max, int\*\* arr, int rows, int cols, int\* countRM, int\* countCM)
* Данная функция получает в качества параметров массив arr3 структуры index, в поля каждого из элементов данного массива будет записываться адрес максимального элемента. Переменная Max, которая отвечает за хранение значения максимального элемента двумерного динамического массива. Сам двумерный динамический массив arr. Переменную rows, cols, которые отвечают за количество строк и столбцов в двумерном динамическом массиве arr. И переменные countRM, countCM. Данные переменные отвечают за подсчет количества строк и столбцов в которых находятся максимальные элементы.
* В начале данная функция проходится по двумерному динамическому массиву с помощью двух циклов for. И в случае если считываемый элемент равен Max, то номер строки и столбца записывается в соответствующие поля массива arr3.

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

{

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

{

if (arr[i][j] == Max)

{

arr3[c].countR = i;

arr3[c].countC = j;

c++;

}

}

}

* Далее создается два одномерных динамических массива n1 и n2 и заполняются значениями из полей массива arr3.

int\* n1 = new int[c];

int\* n2 = new int[c];

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

n1[i] = arr3[i].countC;

n2[i] = arr3[i].countR;

}

* Далее два этих массива сортируются с помощью функции sort(), далее через цикл for проходится функция проходится по элементам массивов и сравнивается с предыдущим, для того чтобы посчитать и отсеять совпадающие строки и столбцы. И если строки и столбцы отличаются, то мы увеличиваем переменные countCM, countRM.

sort(n1, n1 + c);

sort(n2, n2 + c);

for (int i = 1; i < c; i++) {

if (n1[i] != n1[i - 1]) (\*countCM)++;

if (n2[i] != n2[i - 1]) (\*countRM)++;

}

* Полный код функции void Search(index\* arr3, int Max, int\*\* arr, int rows, int cols, int\* countRM, int\* countCM);

void Search(index\* arr3, int Max, int\*\* arr, int rows, int cols, int\* countRM, int\* countCM)

{

int c = 0;

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

{

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

{

if (arr[i][j] == Max)

{

arr3[c].countR = i;

arr3[c].countC = j;

c++;

}

}

}

int\* n1 = new int[c];

int\* n2 = new int[c];

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

n1[i] = arr3[i].countC;

n2[i] = arr3[i].countR;

}

sort(n1, n1 + c);

sort(n2, n2 + c);

for (int i = 1; i < c; i++) {

if (n1[i] != n1[i - 1]) (\*countCM)++;

if (n2[i] != n2[i - 1]) (\*countRM)++;

}

}

* 1. void Del(int\*\* arr, int\* rows, int\* cols, int\*\* arr2, int\* rows2, int\* cols2, int\* count, index\* arr3)
* Данная функция принимает в качестве динамический массив указателей arr. Переменные rows и cols которые хранят количество строк и столбцов в массиве arr. Так же передается новый двумерный динамический массив arr2. А также переменные cols2, rows2, которые хранят размер нового динамического массива. Ну и конечно массив arr3.
* Функция проходится по всем переменным массива arr и проверяет находятся ли они в строках или столбцах, в которых находятся максимальные элементы. И в случае, если считанный элемент не находится в этих строках или столбцах, то мы копируем данный элемент в массив arr2.

void Del(int\*\* arr, int\* rows, int\* cols, int\*\* arr2, int\* rows2, int\* cols2, int\* count, index\* arr3)

{

int q = 0;

int w = 0;

bool f = 0;

for (int i = 0; i < \*rows; i++)

{

w = 0;

for (int j = 0; j < \*cols; j++)

{

for (int k = 0; k < \*count; k++)

{

if (i != arr3[k].countR && j != arr3[k].countC)

{

f = false;

}

else

{

f = true;

break;

}

}

if (f == false)

{

arr2[q][w] = arr[i][j];

w++;

}

}

if (w != 0) {

q++;

}

}

}

* 1. int main()
* Это главная функция программы, в которой происходит инициализация большинства переменных. Создаются двумерные динамические массивы.

int rows = 0;

int cols = 0;

int count = 1;

int Max = -1;

int countRM = 1;

int countCM = 1;

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

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

arr[i] = new int[cols];

}

* Так же вызываются все остальные функции.

FillArr(arr, &rows, &cols);

ShowArr(arr, &rows, &cols);

countMax(arr, &rows, &cols, &count, &Max);

Search(arr3, Max, arr, rows, cols, &countRM, &countCM);

Del(arr, &rows, &cols, arr2, &rows2, &cols2, &count, arr3);

ShowArr(arr2, &rows2, &cols2);

* А так же удаляются все созданные динамические массивы.

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

delete[] arr[i];

}

delete[] arr;

for (int i = 0; i < rows2; i++) {

delete[] arr2[i];

}

delete[] arr2;

delete[] arr3;

* Полный код фукнции main()

int main()

{

setlocale(LC\_ALL, "Ru");

srand(time(0));

int rows = 0;

int cols = 0;

int count = 1;

int Max = -1;

int countRM = 1;

int countCM = 1;

countRC(&rows, &cols);

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

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

arr[i] = new int[cols];

}

FillArr(arr, &rows, &cols);

ShowArr(arr, &rows, &cols);

countMax(arr, &rows, &cols, &count, &Max);

index\* arr3 = new index[count];

Search(arr3, Max, arr, rows, cols, &countRM, &countCM);

int rows2 = rows - countRM;

int cols2 = cols - countCM;

if (rows2 == 0 || cols == 0) {

cout << endl << "В массиве не осталось элементов!" << endl;

return 0;

}

int\*\* arr2 = new int\* [rows2];

for (int i = 0; i < rows2; i++) {

arr2[i] = new int[cols2];

}

Del(arr, &rows, &cols, arr2, &rows2, &cols2, &count, arr3);

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

delete[] arr[i];

}

delete[] arr;

ShowArr(arr2, &rows2, &cols2);

for (int i = 0; i < rows2; i++) {

delete[] arr2[i];

}

delete[] arr2;

delete[] arr3;

return 0;

}

**Полный код**

#include <iostream>

#include <string>

#include <ctime>

#include <algorithm>

using namespace std;

struct index

{

int countR;

int countC;

};

void countRC(int\* rows, int\* cols)

{

while (\*rows <= 0)

{

cout << "Введите количество строк: ";

cin >> \*rows;

}

while (\*cols <= 0)

{

cout << "Введите количество столбцов: ";

cin >> \*cols;

}

}

void FillArr(int\*\* arr, int\* rows, int\* cols)

{

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

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

arr[i][j] = rand() % 20;

}

}

}

void ShowArr(int\*\* arr, int\* rows, int\* cols)

{

cout << endl;

for (int i = 0; i < \*rows; i++)

{

for (int j = 0; j < \*cols; j++)

{

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

}

cout << endl;

}

}

void countMax(int\*\* arr, int\* rows, int\* cols, int\* count, int\* Max)

{

for (int i = 0; i < \*rows; i++)

{

for (int j = 0; j < \*cols; j++)

{

if (\*Max <= arr[i][j])

{

if (\*Max == arr[i][j])

{

(\*count)++;

}

else

{

\*Max = arr[i][j];

\*count = 1;

}

}

}

}

}

void Search(index\* arr3, int Max, int\*\* arr, int rows, int cols, int\* countRM, int\* countCM)

{

int c = 0;

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

{

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

{

if (arr[i][j] == Max)

{

arr3[c].countR = i;

arr3[c].countC = j;

c++;

}

}

}

int\* n1 = new int[c];

int\* n2 = new int[c];

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

n1[i] = arr3[i].countC;

n2[i] = arr3[i].countR;

}

sort(n1, n1 + c);

sort(n2, n2 + c);

for (int i = 1; i < c; i++) {

if (n1[i] != n1[i - 1]) (\*countCM)++;

if (n2[i] != n2[i - 1]) (\*countRM)++;

}

}

void Del(int\*\* arr, int\* rows, int\* cols, int\*\* arr2, int\* rows2, int\* cols2, int\* count, index\* arr3)

{

int q = 0;

int w = 0;

bool f = 0;

for (int i = 0; i < \*rows; i++)

{

w = 0;

for (int j = 0; j < \*cols; j++)

{

for (int k = 0; k < \*count; k++)

{

if (i != arr3[k].countR && j != arr3[k].countC)

{

f = false;

}

else

{

f = true;

break;

}

}

if (f == false)

{

arr2[q][w] = arr[i][j];

w++;

}

}

if (w != 0) {

q++;

}

}

}

int main()

{

setlocale(LC\_ALL, "Ru");

srand(time(0));

int rows = 0;

int cols = 0;

int count = 1;

int Max = -1;

int countRM = 1;

int countCM = 1;

countRC(&rows, &cols);

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

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

arr[i] = new int[cols];

}

FillArr(arr, &rows, &cols);

ShowArr(arr, &rows, &cols);

countMax(arr, &rows, &cols, &count, &Max);

index\* arr3 = new index[count];

Search(arr3, Max, arr, rows, cols, &countRM, &countCM);

int rows2 = rows - countRM;

int cols2 = cols - countCM;

if (rows2 == 0 || cols == 0) {

cout << endl << "В массиве не осталось элементов!" << endl;

return 0;

}

int\*\* arr2 = new int\* [rows2];

for (int i = 0; i < rows2; i++) {

arr2[i] = new int[cols2];

}

Del(arr, &rows, &cols, arr2, &rows2, &cols2, &count, arr3);

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

delete[] arr[i];

}

delete[] arr;

ShowArr(arr2, &rows2, &cols2);

for (int i = 0; i < rows2; i++) {

delete[] arr2[i];

}

delete[] arr2;

delete[] arr3;

return 0;

}















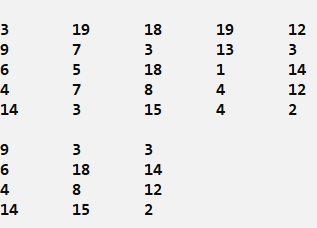


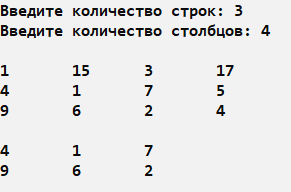
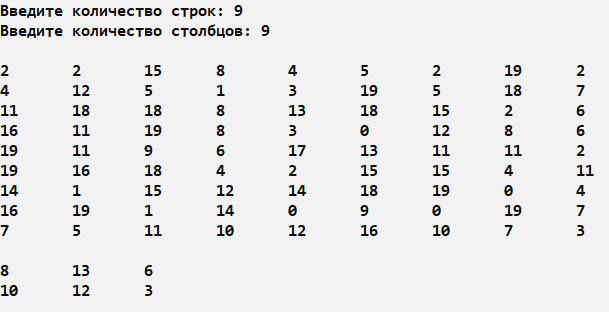
**Скриншоты результатов работы программы**

1. **Нормальный ввод:**
   1. Программа просит ввести количество строк и столбцов. Пример (5 и 5)

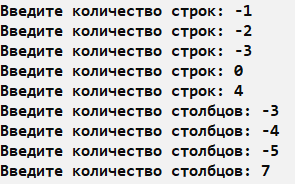
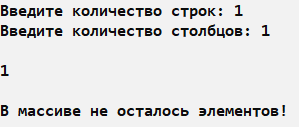


* 1. Программа генерирует двумерный динамический массив. И удаляет все строки и столбцы в которых находятся максимальные элементы в данном случае 19.



* 1. Еще несколько примеров выполнения работы программы.
* 
* 

1. Неправильный ввод:
   1. Попробуем ввести отрицательные числа в значения количества строк и столбцов. Программа не дает ввести отрицательное значение данных переменных.

* 
  1. Попробуем ввести двумерный динамический массив размеров 1 на 1. Программа выведет сообщение о том что все элементы были удалены, так как в изначальном массиве был 1 элемент.
* 
  1. Еще несколько скриншотов работы программы.
* 