МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Инженерно-экономический факультет

Кафедра экономики

Отчет

по лабораторной работе №5

на тему «УКАЗАТЕЛИ. ПРОГРАММИРОВАНИЕ  
С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКИХ  
ДВУМЕРНЫХ МАССИВОВ»

по курсу «Основы алгоритмизации и программирования»

Вариант №7

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Студент гр. 274003  Жерко В.A. |
| Проверила: |  | Петрович Ю.Ю. |

Минск 2022

**Цель:** изучение указателей и распределения динамической памяти

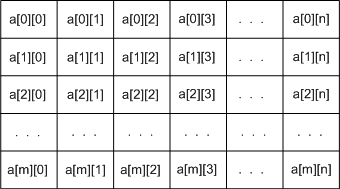
**Краткие теоретические сведения:**

Массив ­- это структура данных, представленная в виде группы ячеек одного типа, объединенных под одним единым именем. Массивы используются для обработки большого количества однотипных данных. Отдельная ячейка данных массива называется элементом массива. Элементами массива могут быть  данные любого типа. Массивы могут иметь как одно, так и более одного измерений. В зависимости от количества измерений массивы делятся на одномерные массивы, двумерные массивы, трёхмерные массивы и так далее до n-мерного массива. Чаще всего в программировании используются одномерные и двумерные массивы, поэтому мы рассмотрим только эти  массивы.

Массивы бывают статические и динамические. Разница между ними в том, что размер статического массива программа знает заранее, а не узнает во время выполнения программы. Во время динамического распределения выделяется память вручную, именно поэтому он и динамический.

**Двумерные массивы в С++**

В двумерном массиве, кроме количества элементов массива, есть такие характеристики как, количество строк и количество столбцов двумерного массива. То есть, визуально, двумерный массив — это обычная таблица, со строками и столбцами. Фактически двумерный массив — это одномерный массив одномерных массивов.



int mas[5][7] = {}; Где 5 – кол-во строк, 7 – кол-во столбцов.

Размер массива (кол-во элементов) считается путём умножения количества строк на столбцы.

**Указатели в С++**

Указатели представляют собой объекты, значением которых служат адреса других объектов (переменных, констант, указателей) или функций. Как и ссылки, указатели применяются для косвенного доступа к объекту. Однако в отличие от ссылок указатели обладают большими возможностями.

Для определения указателя надо указать тип объекта, на который указывает указатель, и символ звездочки \*. Например, определим указатель на объект типа int: **int** \*p;

Указатели, как правило, используются при работе с динамической памятью

В языке C++ для выделения и освобождения памяти определены операции  
new и delete.

Оператор new используется в следующих формах:

* **new тип;** — для переменных
* **new тип[размер];** — для массивов

Освобождение памяти связано с тем, как выделялась память – для одного элемента или для нескольких. В соответствии с этим существует и две формы применения delete:

* **delete указатель;** — для одного элемента
* **delete[] указатель;** — для массивов

**Индивидуальное задание (Вариант 7):**

Дана матрица размером NxM. Упорядочить ее строки по возрастанию их  
первых элементов.

**Код программы:**

#include <iostream>

using namespace std;

int main()

{

char choice[80];

bool des = true;

do {

int stroka, stolb;

cout << "Stroka(N) : "; cin >> stroka; cout << endl;

cout << "Stolb(M): "; cin >> stolb; cout << endl;

cout << "Your array : " << stroka << "\*" << stolb << ":::" << endl;

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

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

arr[i] = new int[stolb];

}

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

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

arr[i][j] = rand() % 100 - 20;

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

}

cout << endl;

}

cout << "Sorted array for the stroki : " << endl;

int min = 0;

int counter = 0;

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

int min = i;

for (int i = counter; i < stroka; i++){

if (arr[min][0] > arr[i][0]){

min = i;

}

}

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

swap(arr[counter][j], arr[min][j]);

}

counter++;

}

cout << endl;

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

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

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

}

cout << endl;

}

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

delete[] arr[i];

}

delete[] arr;

cout << "Want to continue ? Yes(1) No(0) : ";

point :

cin >> choice;

if (strlen(choice) >= 2){

cout << "Wrong number, try again... \n";

goto point;

}

switch (\*choice){

case '1': des = true; break;

case '0': des = false; break;

default: cout << "Wrong number, try again ... \n"; goto point;

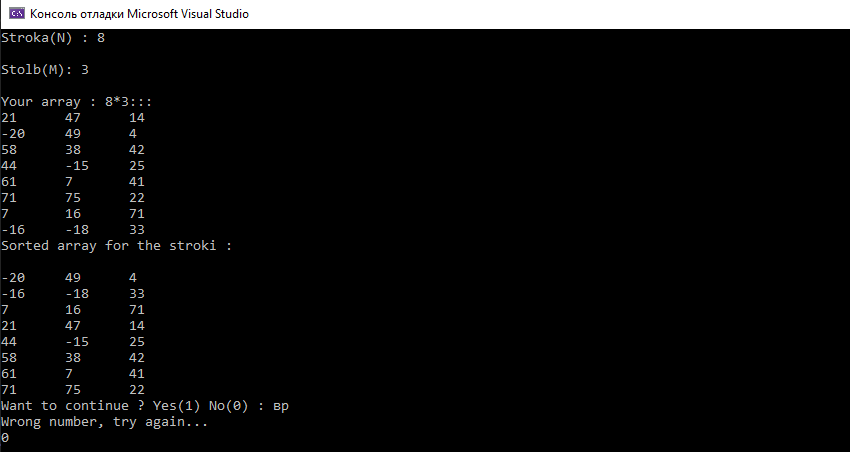
}

} while (des);

return 0;

}

Вывод результата в консоль:



**Вывод:**

В пятой лабораторной работе разобрался, как выделять и освобождать динамическую память при помощи операторов new(delete) и указателей на тип переменной. В этой лабораторной работе указатели использовались для выделения памяти, а именно для создания двумерного динамического массива (Размер вводит пользователь).