МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ»

КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

Лабораторная работа №3

по дисциплине «Основы программирования»

Факультет: ПМИ

Группа: ПМИ-03

Студенты: Сидоров Д.И., Малыгин С. А.

Преподаватель: Еланцева Е.Л.

НОВОСИБИРСК

2020

1. *Условие задачи*: В матрице порядка N поменять местами строки, содержащие минимальный и максимальный элемент. Если минимальный и максимальный элементы принадлежат одной строке, то поменять местами соответствующие столбцы.
2. *Анализ данных*:

*Входные данные*: N > 1, матрица NхN – состоящая из целых чисел.

*Выходные данные*: Изменённая матрица, min, max.

*Метод решения*: Запустим цикл и будем вводить значения матрицы( предварительно введя её размерность N) и, по мере заполнения матрицы, будем искать максимальный и минимальный элемент. Как только найдём – зафиксируем номера строк и столбцов минимального и максимального элемента. Потом запустим цикл в котором будем менять местами значения строки, содержащей минимальный элемент, и значения строки, содержащей максимальный элемент. Если минимальный и максимальный элементы принадлежат одной строке, то запустим цикл в котором будем менять местами значения столбцов, содержащих максимальный и минимальный элемент. После чего выведем изменённую матрицу.

3) *Алгоритм:*

Вывод “Введите N”

Ввод N

int\*\* MX = new int\*[N];

i =0;i < N;

i++

0

1

MX[i] = new int[N+1];

i =0;i < N;

i++

1

j =0;j < N;

j++

0

1

Ввод MX[i][j]

MX[i][j] > max

0

1

max = MX[i][j]

astr = i;

asto = j;

MX[i][j] < min

0

1

max = MX[i][j]

astr = i;

asto = j;

I=0;i < N;i++

0

j= 0;j< N;j++

1

0

1

MX[i][j]

Вывод: min

Вывод: max

astr == bstr

i < N

j < N

memory = MX[i][asto];

MX[i][asto] = MX[i][bsto];

MX[i][bsto] = memory;

memory = MX[astr][j];

MX[astr][j] = MX[bstr][j];

MX[bstr][j] = memory;

1

0

0

0

1

1

i=0;i < N;i++

1

j=0;j < N;j++

0

1

MX[i][j]

i=0;i < N;i++

0

1

delete MX[i];

4)*Текст программы*:

#include<iostream>

#include<time.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(NULL));

int max(0), min(100),N(0),astr(0),bstr(0),asto(0),bsto(0),memory(0);

cout << "Введите N" << endl;

cin >> N;

cout << endl;

int\*\* MX = new int\*[N]; //создаем двумерный массив

for (int i(0); i < N; i++)

{

MX[i] = new int[N+1];

}

for (int i(0); i < N; i++) { // заполняем массив, ищем max и min, запоминаем строку и столбец max и min элемента

for (int j(0); j < N; j++) {

cin >> MX[i][j];

if (MX[i][j] > max) {

max = MX[i][j];

astr = i;

asto = j;

}

if (MX[i][j] < min) {

min = MX[i][j];

bstr = i;

bsto = j;

}

}

}

for (int i(0); i < N; i++) { // выводим матрицу

cout << endl;

for (int j(0); j < N; j++) {

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

}

}

cout << endl << endl;

cout <<"min = " << min << endl; // выводим min элемент

cout <<"max = " << max << endl; // выводим max элемент

if (astr == bstr)

{

for (int i(0); i < N; i++) { // меняем столбцы

memory = MX[i][asto];

MX[i][asto] = MX[i][bsto];

MX[i][bsto] = memory;

}

}

else {

for (int j(0); j < N; j++) { // меняем строки

memory = MX[astr][j];

MX[astr][j] = MX[bstr][j];

MX[bstr][j] = memory;

}

}

for (int i(0); i < N; i++) { // выводим матрицу

cout << endl;

for (int j(0); j < N; j++) {

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

}

}

cout << endl;

for (int i(0); i < N; i++)

{

delete MX[i];

}

delete MX;

}

5) *Тесты*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Дано | | Результат | | Примечание |
| N | Матрица размера N | min  max | Изменённая матрица |
| 1 | 3 | 1 2 3  4 5 0  7 8 9 | min=0  max=9 | 1 2 3  7 8 9  4 5 0 | Минимальный и максимальный элемент в разных строках |
| 2 | 3 | 10 11 12  13 8 4  7 0 20 | min=0  max=20 | 10 12 11  13 4 8  7 20 0 | Минимальный и максимальный элемент в одной строке |
| 3 | 2 | 1 2  3 4 | min=1  max=4 | 3 4  1 2 | Работает при размерности >1 |
| 4 | 3 | 1 2 3  1 4 5  6 7 8 | min=1  max=8 | 6 7 8  1 4 5  1 2 2 | Учитывает первый встреченный минимальный элемент |
| 5 | 3 | 1 2 3  4 5 8  6 7 8 | min=1  max=8 | 4 5 8  1 2 3  6 7 8 | Учитывает первый встреченный максимальный элемент |
|  | | | | | |

6) *Результат работы программы*:

Программа работает правильно, что подтверждают тесты.