

## Лабораторная работа №10

### Поиск максимальных и минимальных элементов в двумерных массивах

#### Оглавление

|  |   |
|--|---|
| Цель .....                                     | 1 |
| Инструкция:.....                               | 2 |
| Задания для самостоятельного выполнения: ..... | 3 |
| Домашнее задание:.....                         | 3 |
| БЛОК А .....                                   | 3 |
| БЛОК В .....                                   | 3 |
| БЛОК С .....                                   | 4 |
| Требования к оформлению программ:.....         | 4 |
| Контрольные вопросы: .....                     | 5 |

#### *Цель*

Научиться находить максимальные и минимальные элементы в матрицах и их позицию.

## ***Инструкция:***

**Задача:** Найти первый максимальный элемент в матрице и его позицию.

## **Реализация:**

```
//-----main.cpp-----  
//Лабораторная работа №10: Поиск максимальных и минимальных элементов в двумерных массивах  
  
#include <iostream>  
#include "matr.h"  
#include "get_matr.h"  
#include "min_max_matr.h"  
  
#define n 5  
#define m 5  
  
#define k 4  
#define p 2  
  
int main()  
{  
    int i,j;  
  
    int **a=new int *[n];  
    for(int i=0;i<n;i++)  
        a[i]=new int [m];  
  
    RandomMatr(a,n,m,-5,5);  
    OutputDescMatr(a,n,m);  
  
    GetMaxMatr(a,n,m,i,j);  
  
    cout<<"Max in matrix: A["<<i+1<<"] ["<<j+1<<"] = "<<a[i][j]<<endl;  
  
    for(int i=0;i<n;i++)  
        delete[] a[i];  
    delete[] a;  
    a=NULL;  
  
    return 0;  
}  
  
#ifndef MIN_MAX_MATR_H_INCLUDED  
#define MIN_MAX_MATR_H_INCLUDED  
  
void GetMaxMatr (int** M,int n,int m,int &i_max,int &j_max);  
  
#endif // MIN_MAX_MATR_H_INCLUDED
```

```

void GetMaxMatr (int** M,int n,int m,int &i_max,int &j_max)
{
    int i,j;
    int max;

    i_max=0;j_max=0;

    max = M[i_max][j_max];

    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            if(max<M[i][j])
            {
                i_max=i;
                j_max=j;
                max=M[i_max][j_max];
            }
        }
    }
}

```

### ***Задания для самостоятельного выполнения:***

1. Использовать Лабораторную № 8. Внести изменения согласно рассмотренному примеру и проанализировать результат.
2. Найти последний максимальный элемент в матрице и его позицию.
3. Найти первый минимальный элемент в матрице и его позицию.

### ***Домашнее задание:***

#### **БЛОК А**

##### ***Задание на отметку «удовлетворительно»***

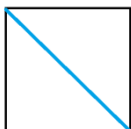
1. Найти максимальный элемент матрицы и его позицию в k-ом столбце.
2. Найти минимальный элемент матрицы и его позицию в k-ой строке.

#### **БЛОК В**

##### ***Задания на отметку «хорошо»***

Найти максимальный элемент и его позицию в заштрихованной области. Элементы матрицы, относящиеся к заштрихованной области, выводить на экран.

1.



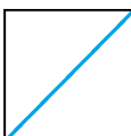
2.



3.



4.



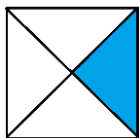
5.



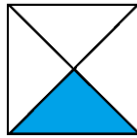
6.



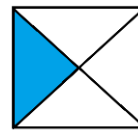
7.



8.

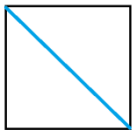


9.



Найти минимальный элемент и его позицию в заштрихованной области. Элементы матрицы, относящиеся к заштрихованной области, выводить на экран.

10.



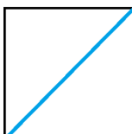
11.



12.



13.



14.



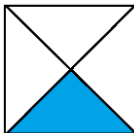
15.



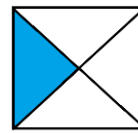
16.



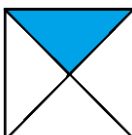
17.



18.



19.



20.



### **БЛОК С**

#### ***Задания на отметку «отлично»***

Для матрицы из своего варианта из *Задания В* найти максимальный отрицательный и минимальный положительный элементы матрицы и их позиции. Элементы матрицы, относящиеся к заштрихованной области, выводить на экран.

#### **Требования к оформлению программ:**

1. **Содержание.** Программа должна делать то, что предусмотрено заданием. Не надо выполнять лишних действий, заданием не предусмотренных.
2. **Спецификация.** В преамбуле программы в комментариях указывать сведения:
  - Кто выполнил.
  - Что делает программа (кратко).
  - Что на входе (имена входных файлов и т.д.).
  - Что на выходе (что является результатов работы программы?).
3. **Ввод и вывод**
  - Приглашения к вводу (например, сколько чисел, какого типа и через какой разделитель нужно вводить).
  - Контрольный вывод (все введенные данные выводить на экран, и только после этого выполнять необходимые вычисления.)

- «Защита от дурака». Проверять вводимые данные на корректность. Например, если необходимо считать количество чего – то, то эта величина не может быть отрицательной и т.д.
4. **Структура кода.** Набираемый код должен быть хорошо структурированным. Использовать:
- Отступы.
  - Комментарии – поясняют решение программы.
  - Осмысленные названия переменных.
  - Пояснения о назначении переменных в комментариях (кроме счетчиков).
5. **Декомпозиция кода**
- Функциональная. Программу оформлять с помощью функций.
6. **Многофайловые проекты**
- Все проекты должны состоять минимум из двух модулей: главного и подключаемого.

Главный модуль **main.cpp**. В нем оставить только функцию `main`, в которой вызывать функции, описанные в других модулях.

Модуль с описанием пользовательских функций **file.cpp** и **file.h**. В этот модуль перенести определение всех функций, необходимых для выполнения задания.

### ***Контрольные вопросы:***

1. Напишите алгоритм нахождения максимального элемента в матрице и его позиции.
2. Напишите функцию для нахождения максимального элемента в матрице и его позиции.
3. Напишите алгоритм нахождения минимального элемента в матрице и его позиции.
4. Напишите функцию для нахождения минимального элемента в матрице и его позиции.