

Лабораторная работа №11

Перестановки в двумерном массиве

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:	5
Контрольные вопросы:	6

Цель

Научиться переставлять элементы в двумерных массивах.

Инструкция:

Задача: Переставить в k-ом столбце матрицы максимальный элемент этого столбца и первый в нем.

Реализация:

```
//-----main.cpp-----
//Лабораторная работа №11: Перестановки в двумерных массивах

#include <iostream>
#include "matr_io.h"
#include "get_matr.h"
#include "min_max_matr.h"
#include "matr_act.h"

#define n 5
#define m 5

using namespace std;

int main()
{
    int i,j,k=1;

    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);

    permutation(a,n,m,k);

    cout<<"Matrix after permutation in "<<k+1<<" column:"<<endl;
    OutputDescMatr(a,n,m);

    for(int i=0;i<n;i++)
        delete[] a[i];
    delete[] a;
    a=NULL;

return 0;
}

//-----matr_act.h-----
#ifndef MATR_ACT_H_INCLUDED
#define MATR_ACT_H_INCLUDED

void permutation(int** A, int N,int M,int k);

#endif // MATR_ACT_H_INCLUDED
```

```

//-----matr_act.cpp-----

#include <iostream>
using namespace std;

void permutation(int** A, int N,int M,int k)
{
    int i,i_max;
    int c,max;

    if (k>=M || k<0)
    {
        cout<<"Error of index k. Elements permutation is impossible "<<endl;
        return;
    }

    i_max=0;
    max=A[i_max][k];

    for(i=0;i<N;i++)
        if(A[i][k]>max)
        {
            i_max=i;
            max=A[i_max][k];
        }

    if(i_max!=0)
    {
        c=A[0][k];
        A[0][k]=A[i_max][k];
        A[i_max][k]=c;
    }
}

```

Задания для самостоятельного выполнения:

1. Использовать Лабораторную № 8. Внести изменения согласно рассмотренному примеру и проанализировать результат.
2. Переставить в k-ой строке матрицы минимальный элемент этой строки и последний в нем.

Домашнее задание:

БЛОК А

Задание на отметку «удовлетворительно»

Переставить местами максимальный и минимальный элемент матрицы. Выводить информацию о переставляемых элементах.

БЛОК В

Задания на отметку «хорошо»

1. Переставить местами первый столбец матрицы и столбец, в котором располагается первый максимальный элемент.
2. Переставить местами первый столбец матрицы и столбец, в котором располагается последний максимальный элемент.
3. Переставить местами первый столбец матрицы и столбец, в котором располагается первый минимальный элемент.
4. Переставить местами первый столбец матрицы и столбец, в котором располагается последний минимальный элемент.
5. Переставить местами первую строку матрицы и строку, в которой располагается первый максимальный элемент.
6. Переставить местами первую строку матрицы и строку, в которой располагается последний максимальный элемент.
7. Переставить местами первую строку матрицы и строку, в которой располагается первый минимальный элемент.
8. Переставить местами первую строку матрицы и строку, в которой располагается последний минимальный элемент.
9. Переставить местами последний столбец матрицы и столбец, в котором располагается первый максимальный элемент.
10. Переставить местами последний столбец матрицы и столбец, в котором располагается последний максимальный элемент.
11. Переставить местами последний столбец матрицы и столбец, в котором располагается первый минимальный элемент.
12. Переставить местами последний столбец матрицы и столбец, в котором располагается последний минимальный элемент.
13. Переставить местами последнюю строку матрицы и строку, в которой располагается первый максимальный элемент.
14. Переставить местами последнюю строку матрицы и строку, в которой располагается последний максимальный элемент.
15. Переставить местами последнюю строку матрицы и строку, в которой располагается первый минимальный элемент.
16. Переставить местами последнюю строку матрицы и строку, в которой располагается последний минимальный элемент.
17. Переставить местами первый столбец матрицы и столбец, в котором больше всего отрицательных элементов.
18. Переставить местами первый столбец матрицы и столбец, в котором больше всего положительных элементов.
19. Переставить местами первую строку матрицы и строку, в которой больше всего отрицательных элементов.
20. Переставить местами первую строку матрицы и строку, в которой больше всего положительных элементов.

БЛОК С

Задания на отметку «отлично»

1. Упорядочить столбцы матрицы по возрастанию сумм элементов в них.
2. Упорядочить столбцы матрицы по убыванию сумм элементов в них.

3. Упорядочить столбцы матрицы по возрастанию среднего арифметического элементов в них.
4. Упорядочить столбцы матрицы по убыванию среднего арифметического элементов в них.
5. Упорядочить столбцы матрицы по возрастанию количества отрицательных элементов в них.
6. Упорядочить столбцы матрицы по убыванию количества отрицательных элементов в них.
7. Упорядочить столбцы матрицы по возрастанию количества нулевых элементов в них.
8. Упорядочить столбцы матрицы по убыванию количества нулевых элементов в них.
9. Упорядочить столбцы матрицы по возрастанию количества положительных элементов в них.
10. Упорядочить столбцы матрицы по убыванию количества положительных элементов в них.
11. Упорядочить строки матрицы по возрастанию сумм элементов в них.
12. Упорядочить строки матрицы по убыванию сумм элементов в них.
13. Упорядочить строки матрицы по возрастанию среднего арифметического элементов в них.
14. Упорядочить строки матрицы по убыванию среднего арифметического элементов в них.
15. Упорядочить строки матрицы по возрастанию количества отрицательных элементов в них.
16. Упорядочить строки матрицы по убыванию количества отрицательных элементов в них.
17. Упорядочить строки матрицы по возрастанию количества нулевых элементов в них.
18. Упорядочить строки матрицы по убыванию количества нулевых элементов в них.
19. Упорядочить строки матрицы по возрастанию количества положительных элементов в них.
20. Упорядочить строки матрицы по убыванию количества положительных элементов в них.

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 перестановки в k-ом столбце матрицы максимального элемента этого столбца и первого в нем.
2. Напишите алгоритм перестановки в k-ой строке матрицы минимального элемента этой строки и последнего в ней.
3. Напишите алгоритм перестановки максимального и минимального элементов матрицы.