

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 7

з дисципліни «Алгоритмізація та програмування»
на тему «Реалізація алгоритмів обробки двовимірних масивів на мові C++»

XAI.301.174.312.8

Виконав студент гр. 312

_____ Діхтяренко М.Г.
(підпис, дата) (П.І.Б.)

Перевірив

_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

2024

МЕТА РОБОТИ

Вивчити теоретичний матеріал з основ представлення двовимірних масивів(матриць) на мові C ++ і реалізувати оголошення, введення з консолі, обробку і виведення в консоль матриць на мові C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на аналіз і виведення елементів матриці.

Введення і виведення даних здійснити в командному вікні.

Matrix40. Дана цілочисельна матриця розміру $M \times N$. Знайти номер останньої з її рядків, що містять максимальну кількість однакових елементів.

Завдання 2. Перетворити матрицю відповідно до свого варіанту завдання, розмір матриці і його елементи ввести з консолі. Вивести результати у консоль.

Matrix50. Дана матриця розміру $M \times N$. Перетворити матрицю, помінявши місцями мінімальний і максимальний елемент в кожному стовпці.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Matrix40.

Вхідні дані (ім'я, опис, тип, обмеження):

- Кількість рядків матриці (rows)
- Кількість стовпців матриці (columns)
- Опис:
 - Функція використовується для створення матриці розміром rows x columns, заповнення її елементами та пошуку кількості парних елементів у кожному рядку.
- Тип даних:
 - int (цілі числа)
- Обмеження:
 - Цілі числа, залежно від розміру типу int.

Алгоритм вирішення показано на рис. 1



Рис.1 - Алгоритм вирішення задачі 1.

Лістинг коду вирішення задачі 1, Matrix40 наведено в дод. А (стор. 6).

Екран роботи програми наведено в дод. Б (стор. 10).

Завдання 2.

Вирішення задачі Matrix50.

Вхідні дані (ім'я, опис, тип, обмеження):

- Кількість рядків матриці (rows)
- Кількість стовпців матриці (columns)
- Опис:
 - Функція також створює матрицю розміром `rows` x `columns`, заповнює її елементами та здійснює редагування так, щоб максимальний елемент замінювався мінімальним у кожному рядку.
- Тип даних:
 - `int` (цілі числа)
- Обмеження:
 - Цілі числа, залежно від розміру типу `int`.

Алгоритм вирішення показано на рис. 2

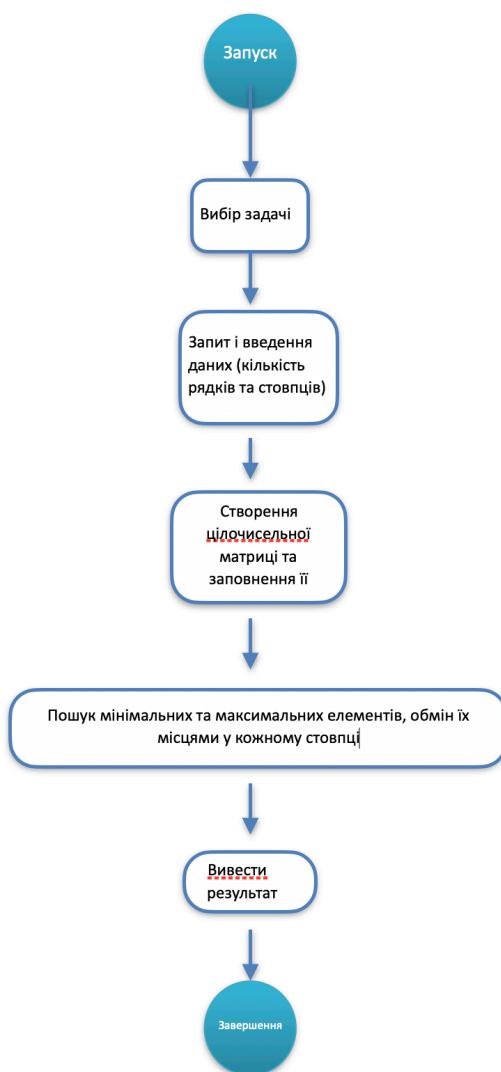


Рис.2 - Алгоритм вирішення задачі 2.

Лістинг коду вирішення задачі 1, Matrix40 наведено в дод. А (стор. 6).
Екран роботи програми наведено в дод. Б (стор. 10).

ВИСНОВКИ

В ході виконання лабораторної роботи 7 було вивчено теоретичний матеріал з основ представлення двовимірних масивів(матриць) на мові C ++ і реалізовано оголошення, введення з консолі, обробку та виведення в консоль матриць на мові C ++ в середовищі Visual Studio, а також відпрацьовані навички написання коду.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#define M 20
#define N 20

using namespace std;

void matrix40();
void expandSize(const int rows, const int columns, int**& arr);
void enterNums(int**&, const int, const int);
void findPaires(int**, const int, const int);
void printm(int rows, int cols, int** a);

void matrix50();
void deleteRow(int**& arr, int& rows, int& cols, const int temp);
void findNums(int**&, const int, const int);

int main() {
    int choice = 0;

    while (choice != 3) {

        cout << "Choose the task: "
             << "\n1.Matrix#40"
             << "\n2.Matrix#50"
             << "\n3.Exit" << endl;

        cin >> choice;

        switch (choice) {
        case 1: {
            matrix40();
            break;
        }
        case 2: {
            matrix50();
            break;
        }
        case 3: {
            cout << "Program is over!";
            break;
        }
        default: {
            cout << "Wrong one, try again!";
        }
    }
}
```

```

}

void matrix40() { /* MATRIX #40*/
    int rows, columns;

    int** arr;

    cout << "Enter rows: ";
    cin >> rows;

    cout << "Enter a columns: ";
    cin >> columns;

    expandSize(rows, columns, arr);
    enterNums(arr, rows, columns);
    printm(rows, columns, arr);

    findPaires(arr, rows, columns);
}

void expandSize(const int rows, const int columns, int**& arr) {
    arr = new int* [rows];
    for (int i = 0; i < rows; i++) {
        arr[i] = new int[columns];
    }
}

void enterNums(int**& arr, const int rows, const int columns) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            cout << "Enter num for arr[" << i << "][" << j << "]: ";
            cin >> arr[i][j];
        }
    }
}

void printm(int rows, int cols, int** a)
{
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++)
            std::cout << a[i][j] << " ";
        std::cout << std::endl;
    }
}

void findPaires(int** arr, const int rows, const int columns) {
    int count = 0, temp = 0, index = 0;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            for (int k = j+1; k < columns; k++) {

```

```

        if (arr[i][j] == arr[i][k]) {
            count++;
            cout << "\nCount is " << count;
        }
    }
    if (count >= temp) {
        index = i;
        temp = count;
    }
    count = 0;
}
cout << "\nINDEX = " << index;
}

```

```

void matrix50() { /*Matrix50*/
    int** arr;
    int rows, cols;

    cout << "Enter rows: ";
    cin >> rows;
    cout << "Enter columns: ";
    cin >> cols;

    expandSize(rows, cols, arr);
    enterNums(arr, rows, cols);
    printm(rows, cols, arr);

    findNums(arr, rows, cols);

    cout << "Array edited\n\n";

    printm(rows, cols, arr);
}

```

```

void findNums(int**& arr, const int rows, const int cols) {

    for (int i = 0; i < rows; i++) {
        int MAX = 3E+5;
        int MIN = 0;
        int MAXtemp = 0;
        int MINtemp = 0;
        for (int j = 0; j < cols; j++) {
            if (arr[i][j] < MAX) {
                MAX = arr[i][j];
                MAXtemp = j;
            }
            if (arr[i][j] > MIN) {
                MIN = arr[i][j];
            }
        }
    }
}

```



```
                MINtemp = j;
            }
        }
        swap(arr[i][MAXtemp], arr[i][MINtemp]);
    }
}
```

ДОДАТОК Б
Скрін-шот вікна виконання програми

```
Choose the task:
1.Matrix#40
2.Matrix#50
3.Exit
1
Enter rows: 3
Enter a columns: 4
Enter num for arr[0][0]: 1
Enter num for arr[0][1]: 22
Enter num for arr[0][2]: 22
Enter num for arr[0][3]: 3
Enter num for arr[1][0]: 4
Enter num for arr[1][1]: 4
Enter num for arr[1][2]: 4
Enter num for arr[1][3]: 1
Enter num for arr[2][0]: 1
Enter num for arr[2][1]: 3
Enter num for arr[2][2]: 1
Enter num for arr[2][3]: 3
1 22 22 3
4 4 4 1
1 3 1 3

Count is 1
Count is 1
Count is 2
Count is 1
Count is 1
Count is 1
INDEX = 1Choose the task:
1.Matrix#40
2.Matrix#50
3.Exit
2
Enter rows: 3
Enter columns: 3
Enter num for arr[0][0]: 1
Enter num for arr[0][1]: 2
Enter num for arr[0][2]: 3
Enter num for arr[1][0]: 1
Enter num for arr[1][1]: 4
Enter num for arr[1][2]: 7
Enter num for arr[2][0]: 2
Enter num for arr[2][1]: 1
Enter num for arr[2][2]: 5
1 2 3
1 4 7
2 1 5
Array edited

3 2 1
7 4 1
2 5 1
Choose the task:
1.Matrix#40
2.Matrix#50
3.Exit
```

Рисунок Б - Екран виконання програми для вирішення завдання 1,2