

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів обходу
масивів»

Варіант 10

Виконав студент ІП-11, Друзенко Олександра Юріївна
(шифр, прізвище, ім'я, по батькові)

Перевірив Мартінова Оксана Петрівна
(прізвище, ім'я, по батькові)

Лабораторна робота 9

Дослідження алгоритмів обходу масивів

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Варіант 10

10	Задано матрицю дійсних чисел $A[m,n]$, ініціалізувати матрицю обходом по стовбцях. У кожному стовпчику матриці знайти останній максимальний елемент і його місцезнаходження. Обміняти знайдене значення X з елементом першого стовбця.
-----------	---

1.Постановка задачі

Ініціалізувати матрицю обходом по стовбцях. Знайти і вивести максимальний елемент стовпчика та його місце. Обміняти максимальний елемент першого стовпця з максимальним елементом інших стовпців

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Обчислення змінної, що описана в п.1, згідно з варіантом.

2.Математична модель

Змінна	Тип	Ім'я	Призначення
Матриця	float	matr	проміжне дане
Кількість рядків	int	m	задане дане
Кількість стовпців	int	n	задане дане
Максимальне значення поточного рядка	float	maxCol	проміжне дане
Максимальне значення серед рядків (окрім 1го)	float	maxCols	проміжне дане
Максимальне значення 1го рядка	float	firstMax	проміжне дане
Масив місцезнаходжень	int	arr	проміжне дане
Функція ініціалізації матриці	function	initMatrix	функція ініціалізації
Функція виведення матриці	function	outputMatrix	функція виведення
Функція обміну максимальних значень	function	maxColumn	функція знаходження та заміни

Функція print() – виведення на екран

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо підпрограму ініціалізації матриці

Крок 3. Деталізуємо підпрограму обміну максимальних значень

Крок 4. Деталізуємо підпрограму виведення матриці в консоль

3.Псевдокод

Крок 1. Основна програма

Початок

```
matr, n, m;  
matr=initMatrix(n,m);  
outputMatrix(matr, n,m);  
maxColumn(matr, m, n);  
outputMatrix(matr, n, m);
```

Кінець

Крок 2. Підпрограма ініціалізації матриці

Початок

Функція initMatrix (matr):

Повторити для j від 0 до m

Повторити для i від 0 до n

$\text{matr}[i][j] = \text{rand}(-10,20);$

Все повторити

Все повторити

Кінець

Крок 3. Підпрограма обміну максимальних значень

Початок

Функція matr(matr,n,m):

$\text{maxCol}, \text{maxCols}=-10, \text{firstMax}=-10, \text{arr}[6]$

Повторити для j від 0 до n

$\text{maxCol} = \text{matr}[0][j];$

Повторити для i від 0 до m

Якщо ($\text{maxCol} \leq \text{matr}[i][j]$)

$\text{maxCol} = \text{matr}[i][j]$

$\text{arr}[2]=i$

$\text{arr}[3]=j$

Все якщо

Все повторити

Якщо ($j == 0$)

$\text{firstMax} = \text{maxCol};$

$\text{arr}[0]=\text{arr}[2]$

$\text{arr}[1]=\text{arr}[3]$

$\text{print}(\text{"max elem"})$

інакше

Якщо (maxCol>maxCols)
 maxCol>maxCols
 arr[4]=arr[2]
 arr[5]=arr[3]

Все якщо
 print(“max elem”)

Все якщо

Все повторити
 matr[arr[4]][arr[5]] = firstMax;
 matr[arr[0]][arr[1]] = maxCols;
 print(“what we change”)

Кінець

Крок 4. Підпрограма виведення матриці в консоль

Початок

Функція outputMatr (matr):

Повторити для j від 0 до m

Повторити для i від 0 до n

print(matr[i][j])

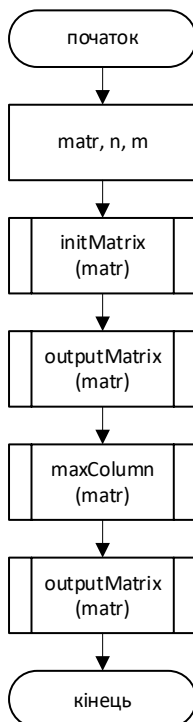
Все повторити

Все повторити

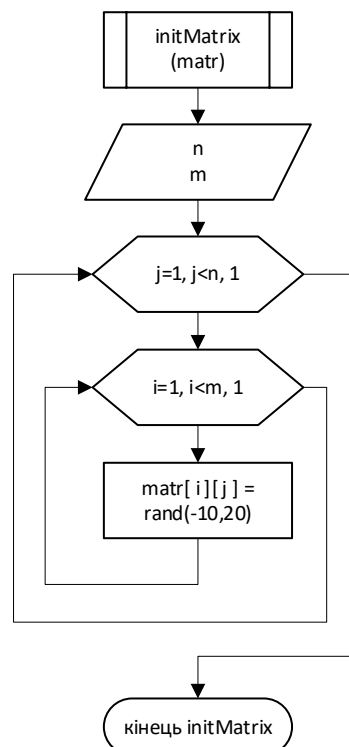
Кінець

4.Блок-схема

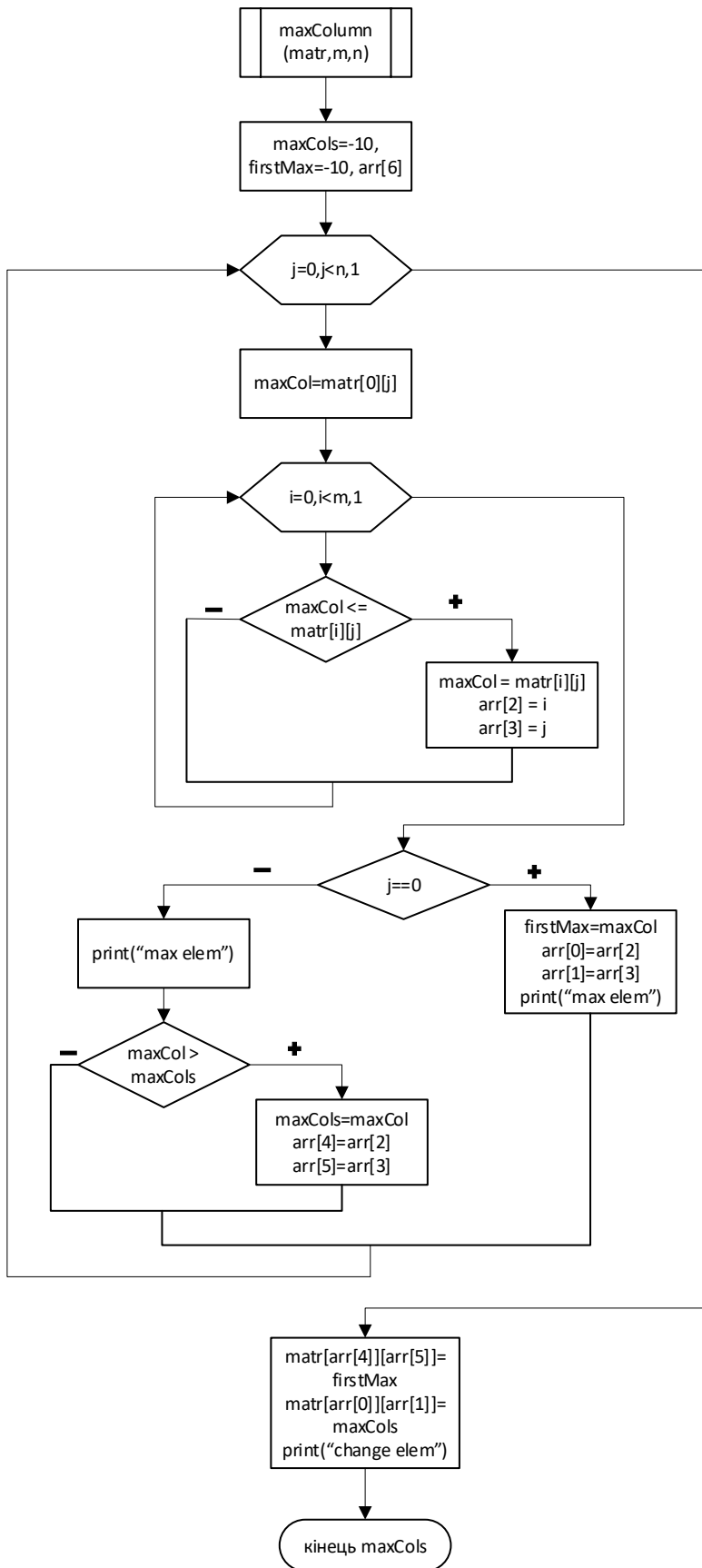
Крок 1



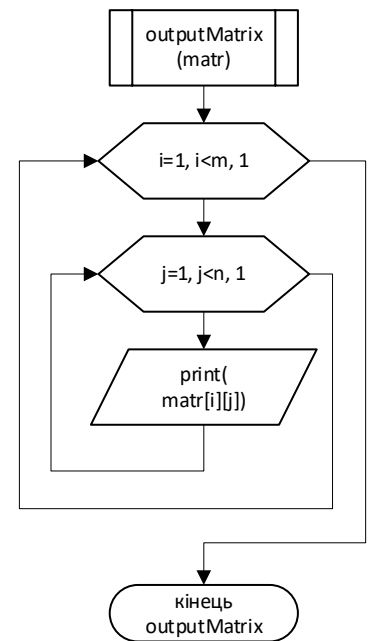
Крок 2. Функція ініціювання матриці



Крок 3. Функція обміну максимальних значень



Крок 4. Функція виведення матриці в консоль



5. Код програми (C++)

```
#include <iostream>
#include <ctime>
#include <stdlib.h>
#include <iomanip>

using namespace std;

float** initMatrix(int&,int&);
void outputMatrix(float**,int,int);
void maxColumn(float**, int, int);

int main()
{
    float** matrixA; int n, m;
    srand(time(NULL));
    matrixA=initMatrix(n,m);
    outputMatrix(matrixA,n,m);
    maxColumn(matrixA, m, n);
    outputMatrix(matrixA, n, m);
}

float** initMatrix(int&n,int&m) {
    cout << "enter m=" ; cin >> m;
    cout << "enter n=" ; cin >> n; cout << endl;
    float** matr = new float* [m];
    for (int i = 0; i < m; i++) {
        matr[i] = new float[n];
    }
    for (int j = 0; j < n; j++) {
        for (int i = 0; i < m; i++) {
            matr[i][j] = rand()%21-10 + (rand()%100/(float)100);
        }
    }
    return matr;
}

void outputMatrix(float**matr,int n, int m) {
    for (int i = 0; i < m; i++) {
        for (int j = 0;j < n; j++) {
            cout << setw(8)<< matr[i][j];
        }
        cout<< endl;
    }
    cout << endl;
}

void maxColumn(float**matr, int m, int n) {
    float maxCol, maxCols=-10, firstMax=-10; int arr[6];
    for (int j = 0; j < n; j++) {
        maxCol = matr[0][j];
        for (int i = 0; i < m; i++) {
            if (maxCol <= matr[i][j]) {
                maxCol = matr[i][j];
                arr[2] = i;
                arr[3] = j;
            }
        }
        if (j == 0) {
            firstMax = maxCol;
            arr[0] = arr[2];
            arr[1] = arr[3];
        }
    }
}
```

```

        cout << "max first element" << " = " << maxCol << " place: [" <<
arr[0]+1 << "]" [" << arr[1]+1 << "]" << endl;
    } else {
        if (maxCol>maxCols){
            maxCols = maxCol;
            arr[4] = arr[2];
            arr[5] = arr[3];
        }
        cout << "max element of column " << arr[3] + 1 << " = " << maxCol
<< " place: [" << arr[2]+1 << "]" [" << arr[3]+1 << "]" << endl;
    }
}
}
matr[arr[4]][arr[5]] = firstMax;
matr[arr[0]][arr[1]] = maxCols;
cout << "we change elements" << " [" << arr[4]+1 << "]" [" << arr[5]+1 <<
"] and [" << arr[0]+1 << "]" [" << arr[1]+1<<"]";
cout << endl << endl;
}

```

6. Тестування програми

```

Microsoft Visual Studio Debug Console
enter m=4
enter n=3

-6.21    4.45    5.16
-8.92    8.77    3.24
-6.9     -8.54   -7.15
-7.22    9.97    -4.58

max first element = -6.21 place: [1] [1]
max element of column 2 = 9.97 place: [4] [2]
max element of column 3 = 5.16 place: [1] [3]
we change elements [4] [2] and [1] [1]

9.97     4.45    5.16
-8.92    8.77    3.24
-6.9     -8.54   -7.15
-7.22    -6.21   -4.58

```

крок	дія
	початок
1	Введення m=4, n=3
2	Матриця: <div> <div>-6.21</div> <div>4.45</div> <div>5.16</div> </div> <div> <div>-8.92</div> <div>8.77</div> <div>3.24</div> </div> <div> <div>-6.9</div> <div>-8.54</div> <div>-7.15</div> </div> <div> <div>-7.22</div> <div>9.97</div> <div>-4.58</div> </div>
3	max first element = -6.21 place: [1] [1] max element of column 2 = 9.97 place: [4] [2] max element of column 3 = 5.16 place: [1] [3]

4	we change elements [4] [2] and [1] [1]
	Матриця результат: <div> <div>9.97</div> <div>4.45</div> <div>5.16</div> </div> <div> <div>-8.92</div> <div>8.77</div> <div>3.24</div> </div> <div> <div>-6.9</div> <div>-8.54</div> <div>-7.15</div> </div> <div> <div>-7.22</div> <div>-6.21</div> <div>-4.58</div> </div>
	кінець

7. Висновок

Отже, сьогодні я дослідила алгоритм обходу масивів, навчилася використовувати їх під час складання програм, ініціалізувала та обходила масив по стовпцях, виводила їх максимальні елементи, та обмінювала значення першого стовпця, з стовпцем максимального елементу.