

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 27

Виконав студент ІП-11 Савенко Олексій Андрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Мартінова О. П.
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №9

Дослідження алгоритмів обходу масивів

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

Варіант 27

Індивідуальне завдання

27	Задано матрицю дійсних чисел $A[m,n]$. При обході матриці по рядках знайти в ній перший максимальний елемент X і його місцезнаходження. Обміняти знайдене значення X з елементом останнього стовбця.
----	---

Постановка задачі

Створити матрицю дійсних чисел A розмірності $[m \times n]$, після цього потрібно знайти максимальний елемент матриці і поміняти його місцями з останнім елементом.

Математична модель

Основна підпрограма

main()

Змінна	Тип	Ім'я	Призначення
Кількість рядків матриці	Цілий	m	Вхідні дані
Кількість стовпців матриці	Цілий	n	Вхідні дані
Матриця	Матриця дійсного типу	A	Проміжні дані, Результат

Підпрограми

Input()

Змінна	Тип	Ім'я	Призначення
Матриця	Матриця дійсного типу	A	Проміжні дані, формальний параметр
Кількість рядків матриці	Цілий	m	Проміжні дані, формальний параметр
Кількість стовпців матриці	Цілий	n	Проміжні дані, формальний параметр
Лічильник арифметичного циклу задля здійснення операцій з масивом	Цілий	i	Проміжні дані
Лічильник арифметичного циклу задля здійснення операцій з масивом	Цілий	j	Проміжні дані

Change()

Змінна	Тип	Ім'я	Призначення
Зміна для отримання значення максимального елемента	Дійсний	max	Проміжні дані
Запам'ятовування рядка максимального елемента	Цілий	pos1	Проміжні дані
Запам'ятовування стовпця максимального елемента	Цілий	pos2	Проміжні дані
Лічильник арифметичного циклу задля здійснення операцій з масивом	Цілий	i	Проміжні дані
Лічильник арифметичного циклу задля	Цілий	j	Проміжні дані

здійснення операцій з масивом			
Матриця	Матриця дійсного типу	A	Проміжні дані, формальний параметр
Кількість рядків матриці	Цілий	m	Проміжні дані, формальний параметр
Кількість стовпців матриці	Цілий	n	Проміжні дані, формальний параметр

Крок 1. Визначимо та охарактеризуємо основні дії алгоритму

Крок 2. Деталізуємо дію Заповнення матриці елементами

Крок 3. Деталізуємо дію Знаходження максимального елемента матриці та заміну з останнім елементом

Псевдокод алгоритму

Крок 1.

Початок

Введення m, n

Деталізуємо дію Заповнення матриці елементами

Деталізуємо дію Знаходження максимального елемента матриці та заміну
з останнім елементом

Кінець

Крок 2.

Початок

Введення m, n

Input(A, m, n)

Деталізуємо дію Знаходження максимального елемента матриці та заміну
з останнім елементом

Кінець

Input(A, m, n)

повторити

для i від 0 до m із кроком 1

повторити

для j від 0 до n із кроком 1

$$A[i][j] = \text{round}(((\text{double})\text{rand}()/3200-5)*100)/100$$

все повторити

все повторити

Кінець

Крок 3.

Початок

Введення m,n

Input(A,m,n)

Change(A,m,n)

Кінець

Input(A, m, n)

повторити

для i від 0 до m із кроком 1

повторити

для j від 0 до n із кроком 1

$A[i][j] = \text{round}(((\text{double})\text{rand}()/3200-5)*100)/100$

все повторити

все повторити

Кінець

Change(A, m, n)

pos1,pos2;

max=A[0][0]

повторити

для i від 0 до m із кроком 1

повторити

для j від 0 до n із кроком 1

якщо max<A[i][j]

то

max = A[i][j]

pos1 =i

pos2=j

якщо j==n-1 && i==m-1

то

A[pos1][pos2]=A[i][j]

A[i][j]=max

все повторити

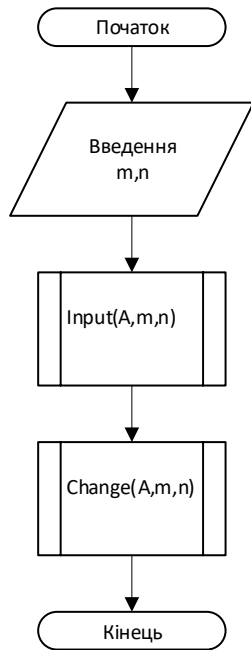
все повторити

Кінець

Блок-схеми алгоритму:

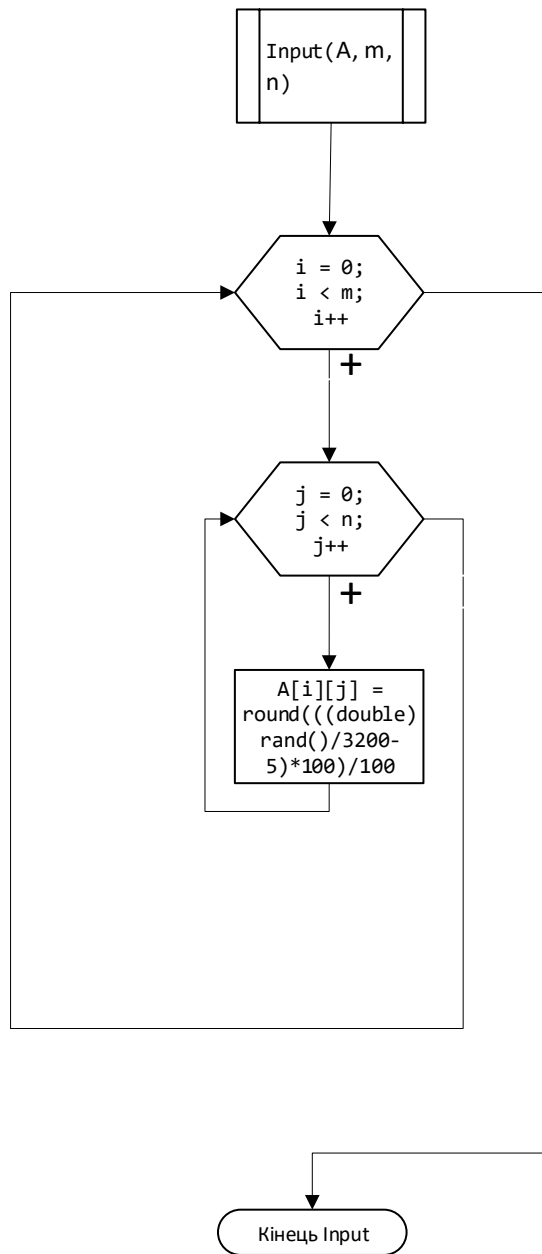
Основна програма

main()

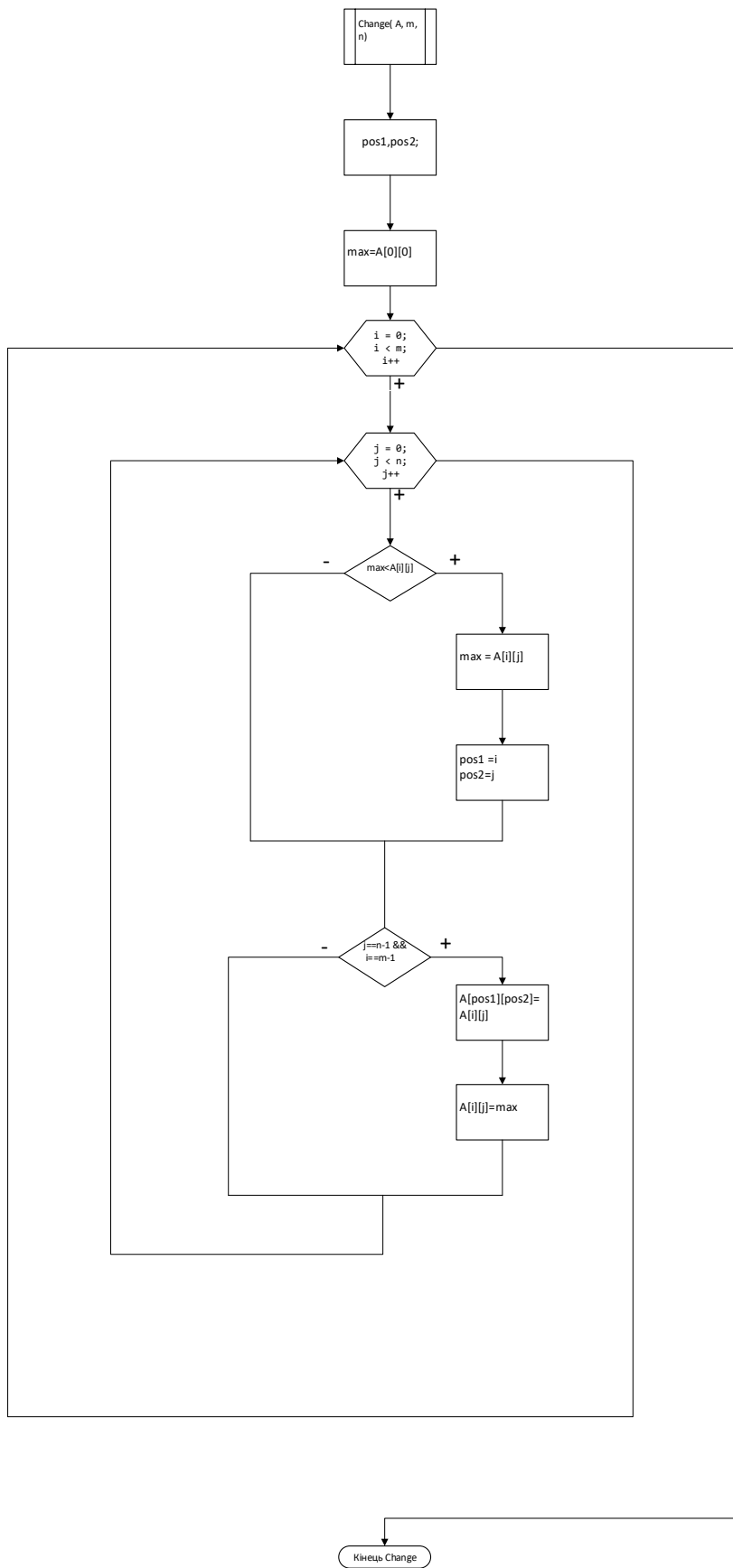


Підпрограми алгоритму

Input()



Change()



Програмний код на мові C++:

```
#include <iostream>
#include <ctime>
using namespace std;

void Generation(double**,int ,int );
void Input(double**,int,int);
void Output(double**,int,int);
void Change(double**,int,int);
void Destruction(double**,int);

int main() {
    srand(time(nullptr));
    int m; cout<<"Enter number of rows: "; cin>>m;
    int n; cout<<"Enter number of pillars: "; cin>>n;
    double** A= new double* [m];

    Generation(A,m,n);
    Input(A,m,n);

    cout<<"Mas A before change"<<endl;
    Output(A,m,n);
    Change(A,m,n);

    cout<<"Mas A before change"<<endl;
    Output(A,m,n);

    Destruction(A,m);
    return 0;
}

void Generation(double**A,int m,int n){
    for(int i=0; i<m;i++){
        A[i]=new double [n];
    }
}

void Input(double**A,int m,int n){
    for(int i =0; i<m;i++){
        for(int j=0;j<n;j++){
            A[i][j]=round(((double)rand())/3200-5)*100)/100;
        }
    }
}

void Output(double** A,int m,int n){
    for(int i = 0; i<m;i++){
        for(int j =0; j<n;j++){
            cout<<A[i][j]<<"\t";
        }
        cout<<endl;
    }
}

void Change(double** A,int m,int n){
    int pos1,pos2;
    double max=A[0][0];
```

```

        for(int i = 0; i<m;i++){

            for(int j =0; j<n;j++){
                if(max<A[i][j]) {
                    max = A[i][j];
                    pos1 =i;
                    pos2=j;
                }
                if(j==n-1 && i==m-1){
                    A[pos1][pos2]=A[i][j];
                    A[i][j]=max;
                }
            }
        }
        cout<<endl;
    }
}

void Destruction(double** A,int m){
    for (int i=0;i<m;i++){
        delete [] A[i];
    }
}

```

Випробування алгоритму на мові програмування C++:

```

"C:\Users\Oleksii Savenko\CLionProjects\untitled17\cmake-build-debug\untitled17.exe"
Enter number of rows: 4
Enter number of pillars: 4
Mas A before change
-4.86   3.78   -3.98   4.17
4.33   -2.53   3.71   0.35
1.1 3.71   4.73   2.13
4.27   2.09   -1.82  -1.18

Mas A after change
-4.86   3.78   -3.98   4.17
4.33   -2.53   3.71   0.35
1.1 3.71  -1.18   2.13
4.27   2.09   -1.82   4.73

```

Випробування алгоритму

Блок	Дія
	Початок
Введення m,n	Enter number of rows: 4; m=4 Enter number of pillars: 4; n=4

1. Input() Заповнення матриці елементами	<p>Mas A before change</p> <p>-4.86 3.78 -3.98 4.17</p> <p>4.33 -2.53 3.71 0.35</p> <p>1.1 3.71 4.73 2.13</p> <p>4.27 2.09 -1.82 -1.18</p>
2. Change() Знаходження максимального елемента матриці та заміна з останнім елементом	<p>max = 4.73</p> <p>pos 1 = i = 2; pos 2 = j=2;</p> <p>A[pos1][pos2]=A[i][j]=-1.18</p> <p>A[i][j]= max = 4.73</p> <p>Mas A after change</p> <p>-4.86 3.78 -3.98 4.17</p> <p>4.33 -2.53 3.71 0.35</p> <p>1.1 3.71 -1.18 2.13</p> <p>4.27 2.09 -1.82 4.73</p>
	<p>Кінець</p>

Висновок

Я дослідив алгоритми обходу масивів, набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій. Була розроблена постановка задачі, де була пояснена логіка алгоритму, математична модель основної програми та підпрограм, псевдокод де розписаний алгоритм, а також блок-схеми основної програми та підпрограм які застосовуються при роботі алгоритму. Для розв'язку заданого завдання було створено та заповнено **Input()** матрицю дійсних чисел **A** розмірності (**m x n**) та видозмінено її за допомогою

перестановки **Change()** максимального елемента та останнього елемента масиву. За допомогою випробування алгоритму, я перевіряв його вірність і довів, що він є вірним.

Отже, мій алгоритм та програмний код написаний на його основі можна використовувати для вирішення завдань даного типу.