

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 22

Виконав студент _____ Мешков Андрій Ігорович _____

(шифр, прізвище, ім'я, по батькові)

Перевірів _____ Вечерковська Анастасія Сергіївна _____

(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 22

Завдання

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

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

22 | Задано матрицю дійсних чисел $A[m,n]$. У кожному стовпчику матриці знайти останній додатний елемент X і його місцезнаходження. Обміняти знайдене значення X з елементом середнього рядка.

Постанова задачі. У консолі вводиться розмір матриці(двовимірний масив). Згенерувати матрицю заданим розміром з випадкових чисел. У кожному стовпцю знайти додатний елемент та його місцезнаходження. Обміняти знайдене значення з елементом середнього рядка.

Результатом буде виведення знайдених значень x , його місцезнаходження та нова матриця.

Побудова математичної моделі: для більшої наочності складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Матриця	Дійсний	A	Проміжні дані, результат
Кількість рядків	Цілочисельний	n	Проміжні дані
Кількість стовпців	Цілочисельний	m	Проміжні дані
Останнє додатне значення	Дійсний	x	Проміжні дані, результат
Параметр циклу	Цілочисельний	i	Проміжні дані
Параметр внутрішнього циклу	Цілочисельний	j	Проміжні дані
Рядок x	Цілочисельний	ix	Проміжні дані, результат

Стовпець x	Цілочисельний	jx	Проміжні дані, результат
Матричний параметр функції	Дійсний	matr	Проміжні дані
Другий параметр функції	Цілочисельний	n0	Проміжні дані
Третій параметр функції	Цілочисельний	m0	Проміжні дані
Середній рядок	Цілочисельний	d	Проміжні дані
Створення матриці	Процедура	input	Початкові дані
Виведення матриці	Процедура	output	Початкові дані
Створення одновимірної масиви	Процедура	last_pos	Початкові дані

Вводимо розмір матриці n , m . Матрицю A згенеруємо за допомогою двох арифметичних циклів випадковими числами всередині функції **input**. Вхідними даними ф-ції є порожня матриця з 10×10 «зарезервованими» місцями A . Одразу виведемо значення матриці у функції **output** по-елементно, використовуючи арифметичний цикл. За допомогою двох змінених місцями арифметичних циклів знайдемо останнє додатне значення x у кожному стовпці у ф-ції **last_pos**, запам'ятовуємо номер рядка ix , номер стовпця jx . Якщо є додатне число, ми змінюємо x зі значенням середнього рядка $matr[d=n/2-1][j]$, діленням націло з округленням до більшого при $i=0,1,2,3\dots$ (у випадку парної кількості рядків середнім вважати перший з двох середніх), виводимо значення x та місцезнаходження ix , jx .

Якщо стовпець немає додатного значення $jx!=j$, виводимо текст: «Стовпець немає додатних чисел» та значення не змінюють місця.

Виведемо змінений масив за допомогою ф-ції **output**.

Виклик всіх функцій відбувається в основній програмі.

Розв'язання. Програмні специфікації запишемо у псевдокоді та у графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію генерації матриці за допомогою підпрограми.

Крок 3. Деталізуємо дію виведення матриці за допомогою підпрограми.

Крок 4. Деталізуємо дію знаходження останнього додатного значення у стовпці та його місцезнаходження.

Крок 5. Деталізуємо дію виводу нової матриці.

Псевдокод

Програма:

Початок

Введення n, m

input(A, n, m)

output(A, n, m)

last_pos(A, n, m)

output(A, n, m)

Кінець

Підпрограми:

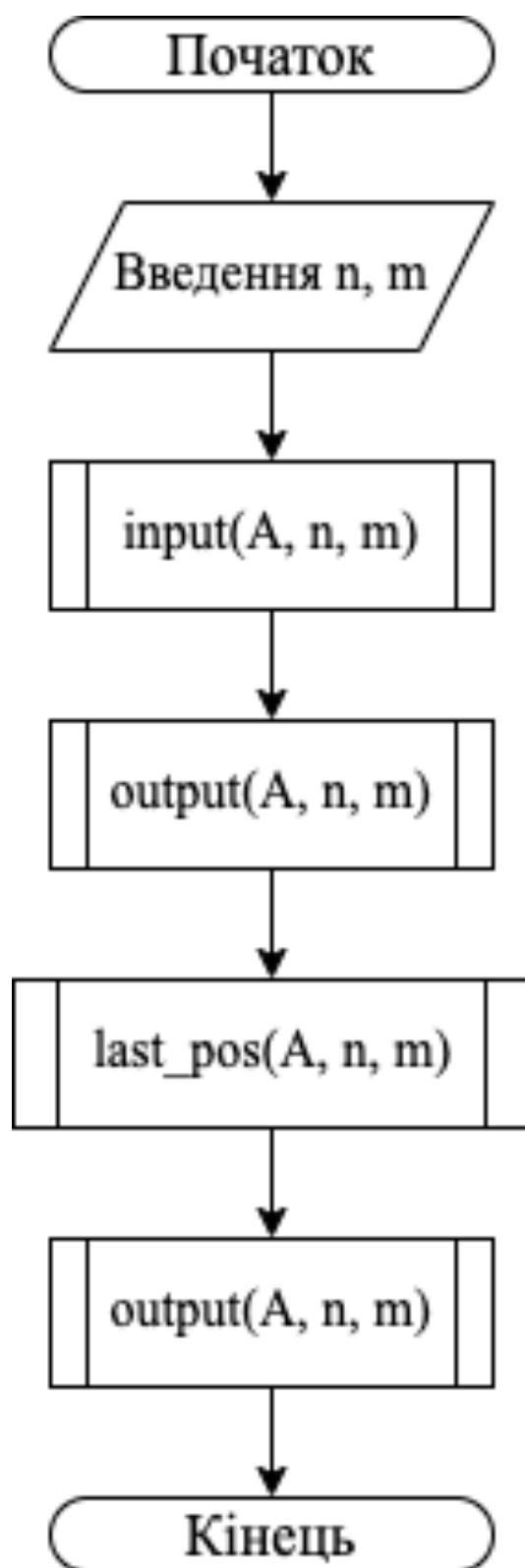
```
input(matr, n0, m0)
    повторити
    для i від 0 до n
        повторити
        для j від 0 до m
            matr[i][j]:=random
        все повторити
    все повторити
кінець
```

```
output(matr, n0, m0)
    повторити
    для i від 0 до n
        повторити
        для j від 0 до m
            Вивести matr[i][j]
        все повторити
    все повторити
кінець
```

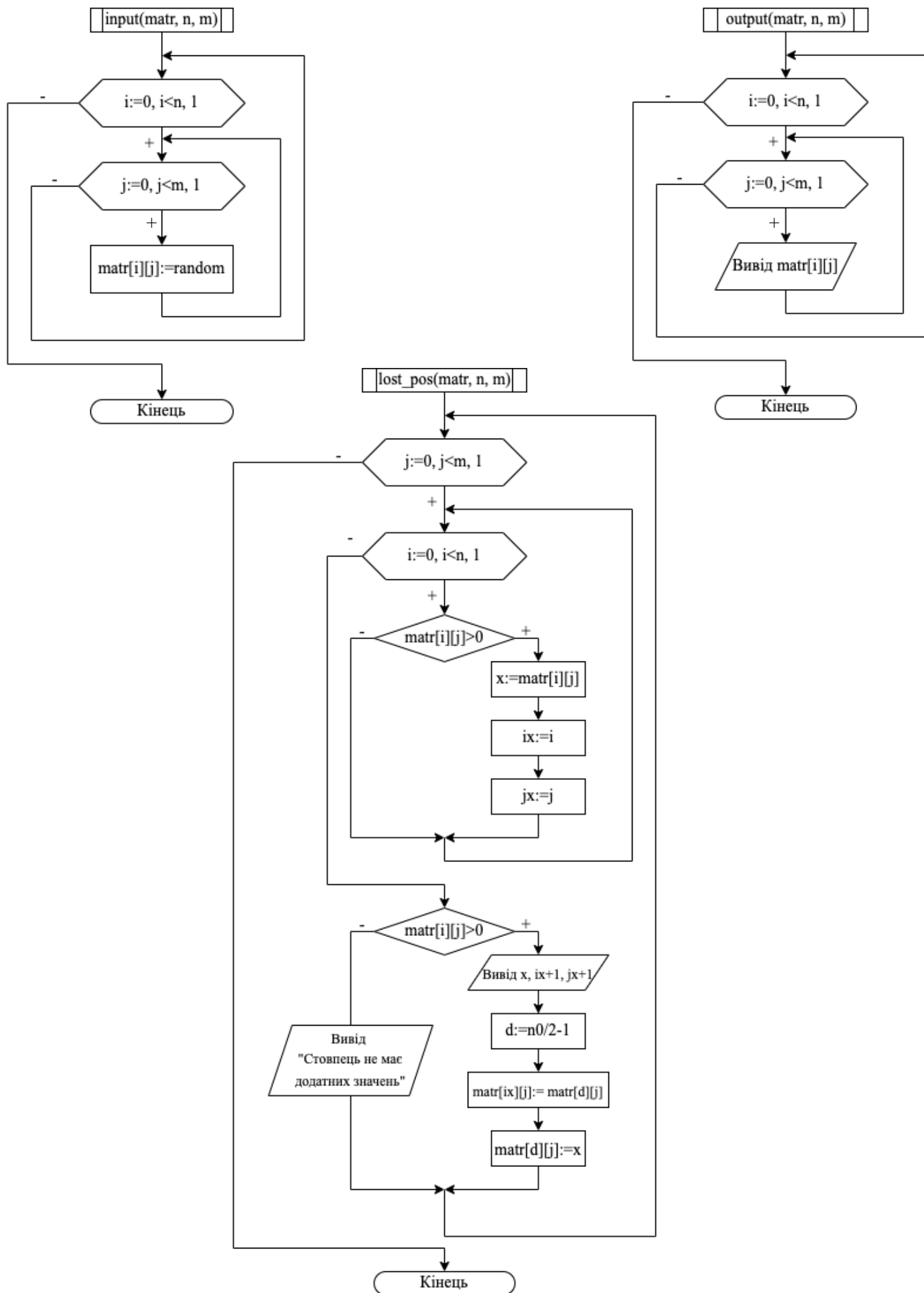
```
last_pos(matr, n0, m0):
    повторити
    для j від 0 до m
        повторити
        для i від 0 до n
            якщо matr[i][j]>0
                то
                x:=matr[i][j]
                ix:=i
                jx:=j
            все якщо
        все повторити
        якщо jx==j
            то
            вивід x, ix+1, jx+1
            d:=n0/2-1
            matr[ix][j]:= matr[d][j]
            matr[d][j]:=x
            інакше
                вивід «Стовпець не має додатних значень»
        все якщо
    все повторити
кінець
```

Блок-схема

Програма:



Підпрограми:



Код програми:

```
1  #include <iostream>
2  #include <iomanip>
3  #include <cstdio>
4  #include <cstdlib>
5  #include <ctime>
6  #include <cmath>
7  using namespace std;
8
9  typedef double Matrix[10][10];
10 Matrix A;
11 void input(Matrix, int, int);
12 void output(Matrix, int, int);
13 void last_pos(Matrix, int, int);
14
15 int main() {
16     int n, m;
17     cout<<"Enter n, m: ";
18     cin >> n >> m;
19     cout << endl << "A(n x m):" << endl;
20     srand(time(NULL)); 2 ⚠ Implicit conversion loses integer precision: 'time_t' (ak
21     input(A, n, m);
22     output(A, n, m);
23     last_pos(A, n, m);
24     output(A, n, m);
25     return 0;
26 }
27
28 void input(Matrix matr, int n0, int m0){
29     int range_min=-200, range_max=200, rang_double=3;
30
31     for (int i=0; i<n0; i++){
32         for (int j=0; j<m0; j++){
33             matr[i][j] = range_min + 1 + rand()%(range_max-range_min + 1) -
34                         (rand()%((int)pow(10,rang_double)) / pow(10,rang_double));
35         }
36     }
37 }
```



```

38 void output(Matrix matr, int n0, int m0){
39     for (int i=0; i<n0; i++){
40         for (int j=0; j<m0; j++){
41             cout<<setw(9)<<matr[i][j];
42         }
43         cout<<endl;
44     }
45     cout<<endl;
46 }
47
48 void last_pos(Matrix matr, int n0, int m0){
49     double x;
50     int d;
51     int ix, jx;
52     for (int j=0; j<m0; j++){
53         for (int i=0; i<n0; i++){
54             if(matr[i][j]>0){
55                 x=matr[i][j];
56                 ix=i;
57                 jx=j;
58             }
59         }
60         if(jx==j){
61             cout<<"x = "<<x<<endl;
62             cout<<"Location: "<<ix+1<<","<<jx+1<<endl<<endl;
63             d=ceil(n0/double(2))-1;
64             matr[ix][j]=matr[d][j];
65             matr[d][j]=x;
66         }
67         else{
68             cout<<"The column "<<j+1<<" does not have a positive number."<<endl<<endl;
69         }
70     }
71 }
72

```

Enter n, m: 5 7

A(n x m):

-14.763	178.976	178.119	-164.996	-115.435	-12.913	-90.633
-168.689	-155.136	74.029	154.036	-128.557	87.657	-170.565
129.192	132.55	-140.282	-120.111	83.361	36.364	-51.12
-31.17	-155.32	8.651	-66.658	87.541	-168.542	-183.385
-49.52	149.35	-91.322	-110.635	88.126	76.376	-185.599

x = 129.192

Location: 3,1

x = 149.35

Location: 5,2

x = 8.651

Location: 4,3

x = 154.036

Location: 2,4

x = 88.126

Location: 5,5

x = 76.376

Location: 5,6

The column 7 does not have a positive number.

-14.763	178.976	178.119	-164.996	-115.435	-12.913	-90.633
-168.689	-155.136	74.029	-120.111	-128.557	87.657	-170.565
129.192	149.35	8.651	154.036	88.126	76.376	-51.12
-31.17	-155.32	-140.282	-66.658	87.541	-168.542	-183.385
-49.52	132.55	-91.322	-110.635	83.361	36.364	-185.599

Program ended with exit code: 0

Enter n, m: 4 6

A(n x m):

170.818	-10.712	152.092	-83.36	-145.6	-42.024
195.477	-41.729	-21.397	38.793	-175.46	-33.665
-93.743	-4.023	64.711	-79.169	78.128	37.972
-62.507	-20.035	149.933	-61.409	43.528	-40.319

x = 195.477

Location: 2,1

The column 2 does not have a positive number.

x = 149.933

Location: 4,3

x = 38.793

Location: 2,4

x = 43.528

Location: 4,5

x = 37.972

Location: 3,6

170.818	-10.712	152.092	-83.36	-145.6	-42.024
195.477	-41.729	149.933	38.793	43.528	37.972
-93.743	-4.023	64.711	-79.169	78.128	-33.665
-62.507	-20.035	-21.397	-61.409	-175.46	-40.319

Program ended with exit code: 0|

Висновок: отже, в результаті виконання лабораторної роботи ми отримали алгоритм для знаходження останнього додатного значення у кожного стовпці та створення матриці за допомогою обміну ост. д. зн. зі значенням середнього рядка. Було досліджено алгоритми обходу масивів, проаналізовано подане завдання, декомпозовано та виконано. Також були розроблені псевдокод, код програми та блок-схема поставленого алгоритму.