

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 № 8 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

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 8

Дослідження алгоритмів пошуку та сортування

Мета – дослідити алгоритми пошуку та сортування, 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. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом (табл. 1).

22	5 x 7	Дійсний	Із середнього арифметичного додатних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.
----	-------	---------	---

Постанова задачі. Згенерувати матрицю(двовимірний масив) розміром 5 x 7. Створити одновимірний масив, значеннями якого буде середнє арифметичне додатних значень елементів кожного стовпця двовимірного масиву. Потім відсортувати масив обміном за спаданням. Результатом буде виведення матриці, початкового та відсортованого одновимірних масивів.

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

Змінна	Тип	Ім'я	Призначення
Матриця	Дійсний	A	Проміжні дані, результат
Одновимірний масив	Дійсний	B	Проміжні дані, результат
Кількість рядків	Цілочисельний	n	Проміжні дані
Кількість стовпців/елементів одновимірного масиву	Цілочисельний	m	Проміжні дані
Змінна для сортування масиву	Цілочисельний	z	Проміжні дані
Крок сортування	Цілочисельний	k	Проміжні дані
Параметр циклу	Цілочисельний	i	Проміжні дані

Параметр внутрішнього циклу	Цілочисельний	j	Проміжні дані
Матричний параметр функції	Дійсний	matr	Проміжні дані
Масивний параметр функції	Дійсний	arr	Проміжні дані
Сума елементів для знаходження сер.-арифм.	Дійсний	a	Проміжні дані
Кількість елементів для знаходження сер.-арифм.	Цілочисельний	s	Проміжні дані
Створення матриці	Процедура	input	Початкові дані
Виведення матриці	Процедура	output	Початкові дані
Створення одновимірного масиву	Процедура	matr_arr	Початкові дані

Матрицю **A** згенеруємо за допомогою двох арифметичних циклів випадковими числами всередині функції **input**. Вхідними даними ф-ції є порожня матриця з 10 x 10 «зарезервованими» місцями **A**. Одразу виведемо значення матриці у функції **output** по-елементно, використовуючи арифметичний цикл. За допомогою двох змінених місцями арифметичних циклів знайдемо значення одновимірного масиву **B** у ф-ції **matr_arr**, у кожному стовпці ми сумуємо значення додатних елементів **a** та їх кількість, знаходимо середнє арифметичне та вставляємо у масив. Виведемо масив **B** за допомогою арифметичного циклу. За допомогою обміну сусідніх елементів масиву, двома циклами відсортуємо масив та виведемо відсортований масив. 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 створення одновимірного масиву за допомогою підпрограми та виведення масиву.

Крок 5. Деталізуємо дію сортування масиву та виведення відсортованого масиву.

Псевдокод

Програма:

Початок

$n := 5$

$m := 7$

input(A, n, m)

output(A, n, m)

matr_arr(A, B, n, m)

повторити

для j від 0 до m

Вивести $B[j]$

все повторити

повторити

для k від 0 до m

повторити

для j від 0 до $m-k$

якщо $B[j] < B[j+1]$

то

$z := B[j]$

$B[j] := B[j+1]$

$B[j+1] := z$

все повторити

все повторити

повторити

для j від 0 до m

Вивести $B[j]$

все повторити

Кінець

Підпрограми:

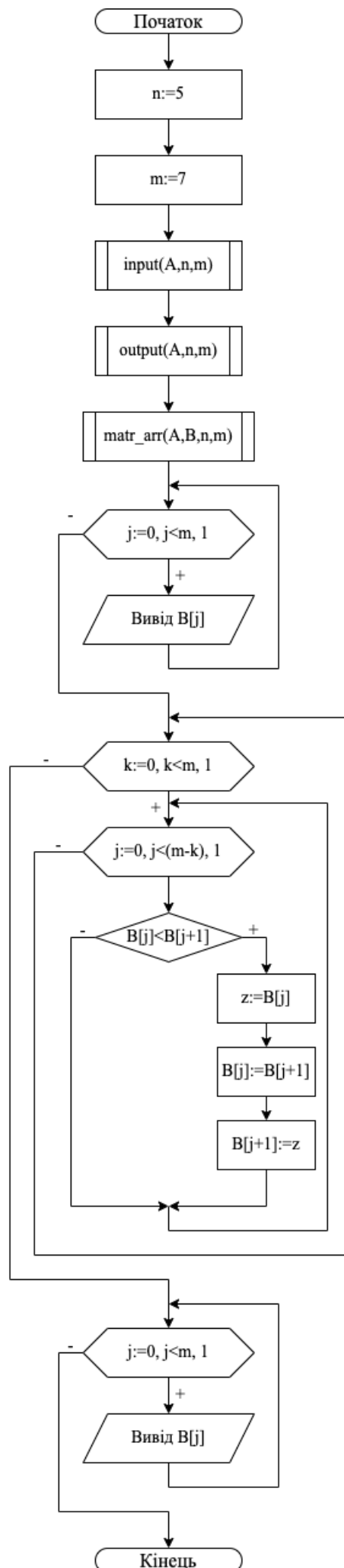
```
input(matr, n, m)
    повторити
    для i від 0 до n
        повторити
        для j від 0 до m
            matr[i][j]:=random
        все повторити
    все повторити
кінець
```

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

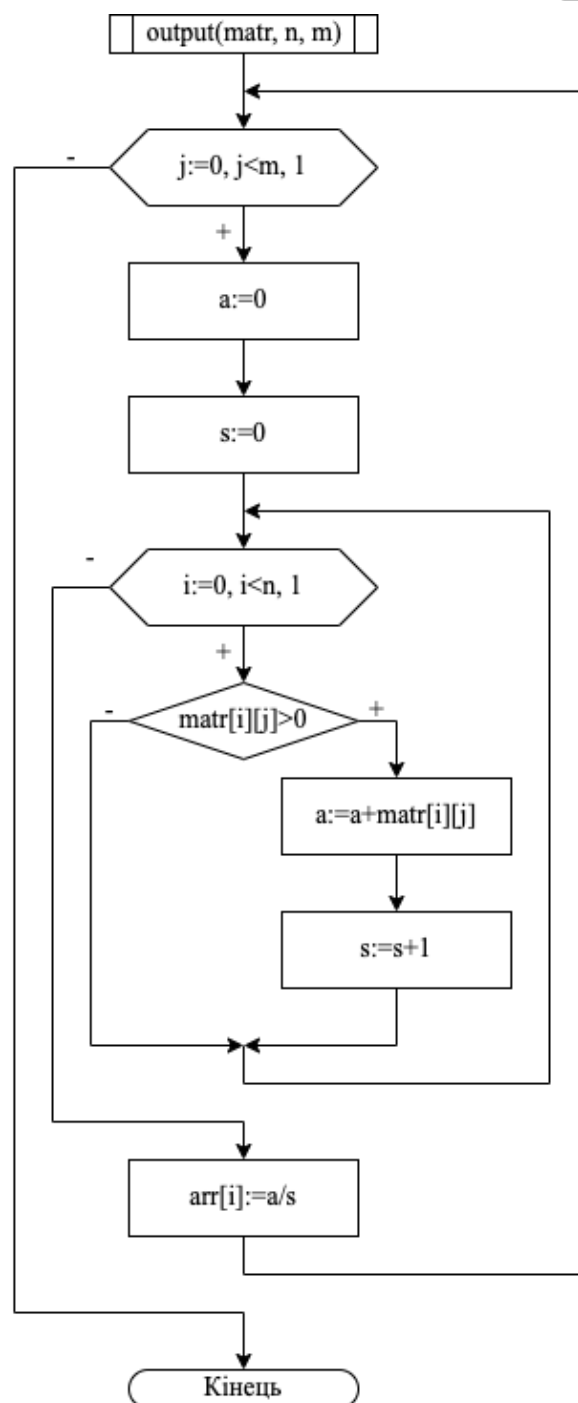
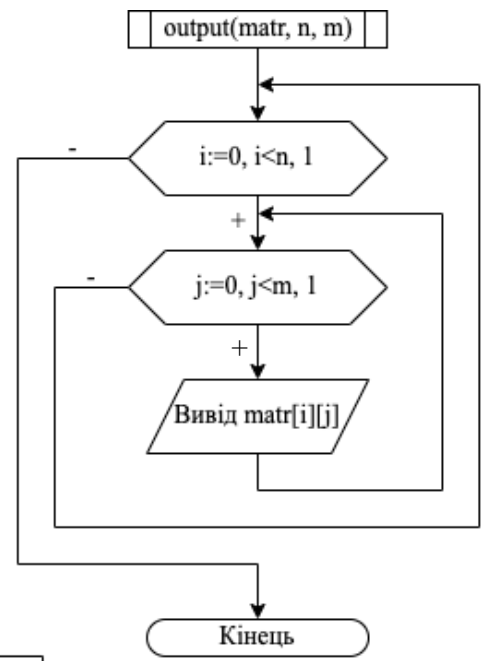
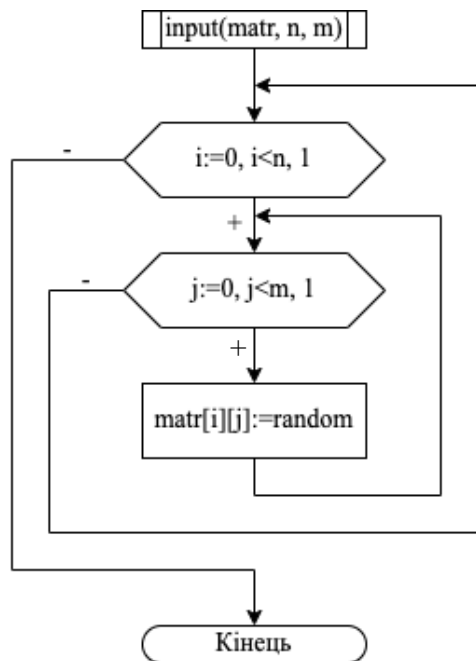
```
matr_arr(matr, arr, n, m):
    повторити
    для j від 0 до m
        a:=0
        s:=0
        повторити
        для i від 0 до n
            якщо matr[i][j]>0
                то
                a:=a+matr[i][j]
                s:=s+1
            все повторити
        arr[j]:=a/s
    все повторити
кінець
```

Блок-схема

Програма:



Підпрограми:



Код програми:

```
1  #include <iomanip>
2  #include <iostream>
3  #include <cstdlib>
4  #include <cstdio>
5  #include <cmath>
6  #include <ctime>
7  using namespace std;
8
9  typedef double Matrix[10][10];
10 Matrix A;
11 typedef double Array[10];
12 Array B;
13 void input(Matrix, int, int);
14 void output(Matrix, int, int);
15 void matr_arr(Matrix, Array, int, int);
16
17 int main() {
18     const int n=5, m=7;
19     double z;
20     srand(time(NULL)); 2 ⚠ Implicit conversion loses integer precision: 'ti
21     cout << "A(n x m):" << endl;
22     input(A, n, m);
23     output(A, n, m);
24     matr_arr(A, B, n, m);
25     cout << "B(m):" << endl;
26     for (int j=0; j<m; j++){
27         cout<<setw(9)<< B[j];
28     }
29     cout<<endl<<endl;
30     cout << "New B(m):" << endl;
31     for(int k; k<m; k++){ 2 ⚠ Variable
32         for (int j=0; j<m-k; j++){
33             if(B[j]<B[j+1]){
34                 z=B[j];
35                 B[j]=B[j+1];
36                 B[j+1]=z;
37             }
38         }
39     }
40     for (int j=0; j<m; j++){
41         cout<<setw(9)<< B[j];
42     }
43     cout<<endl<<endl;
44 }
```



```

45
46 void input(Matrix matr, int n, int m){
47     int range_min=-100, range_max=200, rang_double=3;
48
49     for (int i=0; i<n; i++){
50         for (int j=0; j<m; j++){
51             matr[i][j] = range_min + 1 + rand()%(range_max-range_min + 1) -
                    (rand()%((int)pow(10,rang_double)) / pow(10,rang_double));
52         }
53     }
54 }
55
56 void output(Matrix matr, int n, int m){
57     for (int i=0; i<n; i++){
58         for (int j=0; j<m; j++){
59             cout<<setw(9)<<matr[i][j];
60         }
61         cout<<endl;
62     }
63     cout<<endl;
64 }
65
66 void matr_arr(Matrix matr, Array arr, int n, int m){
67     double a;
68     int s;
69     for (int j=0; j<m; j++){
70         a=0;
71         s=0;
72         for(int i=0; i<n; i++){
73             if(matr[i][j]>0){
74                 a+=matr[i][j];
75                 s++;
76             }
77         }
78         arr[j]=a/double(s);
79     }
80 }
81

```

A(n x m):

67.18	100.279	188.206	-82.27	-74.683	101.007	-39.983
80.068	12.752	182.287	-41.268	60.478	-56.007	48.89
-10.657	-41.31	146.218	107.133	-19.953	82.107	-94.886
120.414	-98.774	11.066	145.643	181.949	82.323	8.489
134.852	131.937	71.258	179.761	167.099	141.446	121.891

B(m):

100.629	81.656	119.807	144.179	136.509	101.721	59.7567
---------	--------	---------	---------	---------	---------	---------

New B(m):

144.179	136.509	119.807	101.721	100.629	81.656	59.7567
---------	---------	---------	---------	---------	--------	---------

Program ended with exit code: 0

Висновок: отже, в результаті виконання лабораторної роботи ми отримали алгоритм для створення масиву на основі середнього арифметичного додатних значень елементів кожного стовпця матриці. Було досліджено алгоритми пошуку та сортування, проаналізовано подане завдання, декомпозовано та виконано. Також були розроблені псевдокод, код програми та блок-схема поставленого алгоритму.