

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 – 18

Виконав студент: Лазьов Кирило Владиславович

Перевірив: Вечерковська Анастасія Сергіївна

Київ 2021

## Лабораторна робота 9

### Дослідження алгоритмів обходу масивів

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

#### Варіант 18

#### Постановка задачі

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

18	Задано матрицю дійсних чисел $A[m,n]$ . При обході матриці по стовпчиках визначити в ній присутність заданого дійсного числа $X$ і його місцезнаходження. Обміняти знайдене значення $X$ з елементом середнього рядка.
----	--

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Дійсний	mat	Вхідні дані
Кількість стовпців	Натуральний	n	Вхідні дані
Кількість рядків	Натуральний	m	Вхідні дані
Шукане число, що задане користувачем	Дійсне	x	Проміжні дані
Номер середнього стовпчика	Натуральний	middle	Проміжні дані
Параметр циклу	Натуральний	i	Проміжні дані
Параметр циклу	Натуральний	j	Проміжні дані
Номер рядка шуканого числа	Натуральний	jx	Результат
Номер стовпчика шуканого числа	Натуральний	ix	Результат

#### Розв'язання

Користувач вводить розмірність матриці: кількість стовпчиків – n і кількість рядків – m. Далі функція matrix\_input генерує матрицю заданої розмірності. Користувач вводить число x і функція xFind шукає його у матриці по

стовпчикам. Коли  $x$  знайдено, він міняється місцями з першим елементом середнього стовпчика. В кінці виводиться змінена матриця.

Крок 1. Визначаємо основні дії

Крок 2. Введення  $n, m$

Крок 3. Виклик першої підпрограми

Крок 4. Введення  $x$

Крок 5. Виклик другої підпрограми

Крок 6. Виведення зміненої функції

## **Псевдокод**

### **Основна програма:**

#### **Початок**

`matr[10][10]`

Введення  $n, m$

$jx = 1, ix = 1$

`matrix_input(matr, m, n)`

Введення  $x$

`x_Find(matr, m, n, x, jx, ix)`

**повторити для  $i$  від 0 до  $m$**

**повторити для  $j$  від 0 до  $n$**

**Виведення `matr[i][j]`**

**Все повторити**

**Все повторити**

**Кінець**

**Підпрограми**

**matrix\_input(double matrix[10][10], int n, int m)**

**Початок**

**повторити для i від 0 до m**

**повторити для j від 0 до n**

**matrix[i][j] = double(rand()) / RAND\_MAX \* 200 + -100**

**все повторити**

**все повторити**

**Кінець**

**xFind**(double matrix[10][10], int m, int n, double X, int jx, int ix)

**Початок**

xz = -1

**повторити для i від 0 до m**

**якщо xz < 0**

**повторити для j від 0 до n**

**якщо x == matrix[j][i]**

**то**

jx = j

ix = i

**все якщо**

**інакше**

**повторити для j від n-1 до 0**

**якщо x == matrix[j][i]**

**то**

jx = j

ix = i

**все якщо**

xz = -xz

**все повторити**

middle = int(n/2) - 1

t = matrix[0][middle]

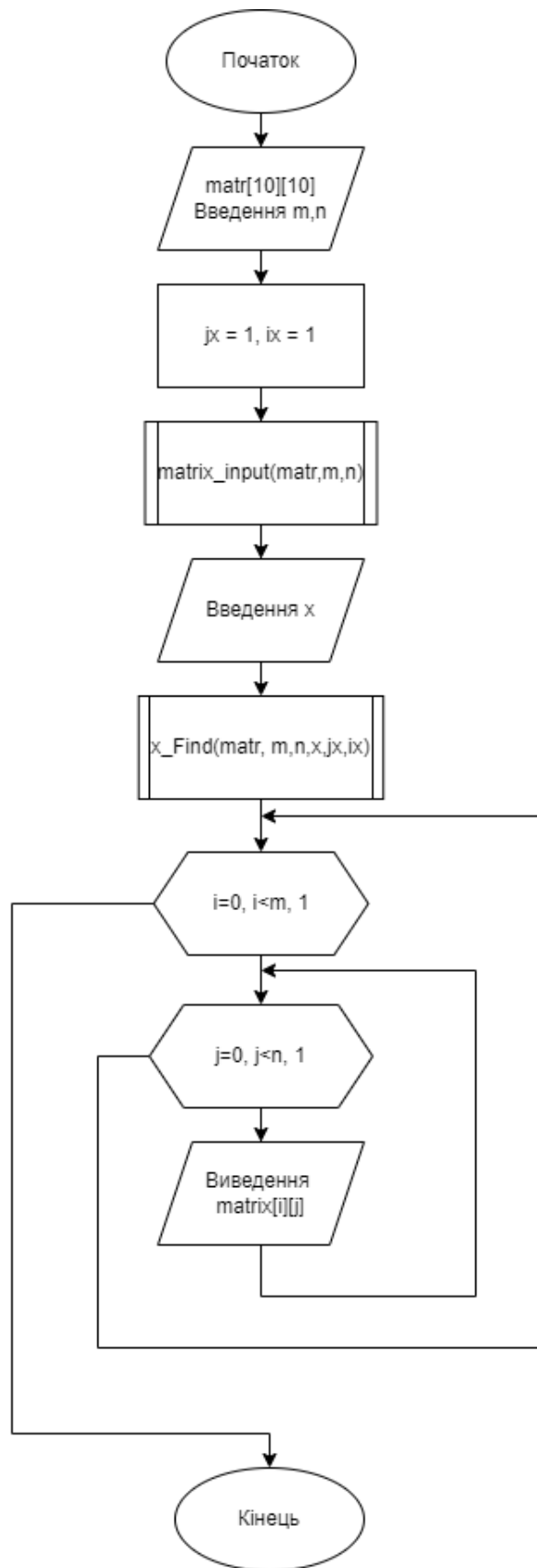
matrix[0][middle] = matrix[jx][ix]

matrix[jx][ix] = t

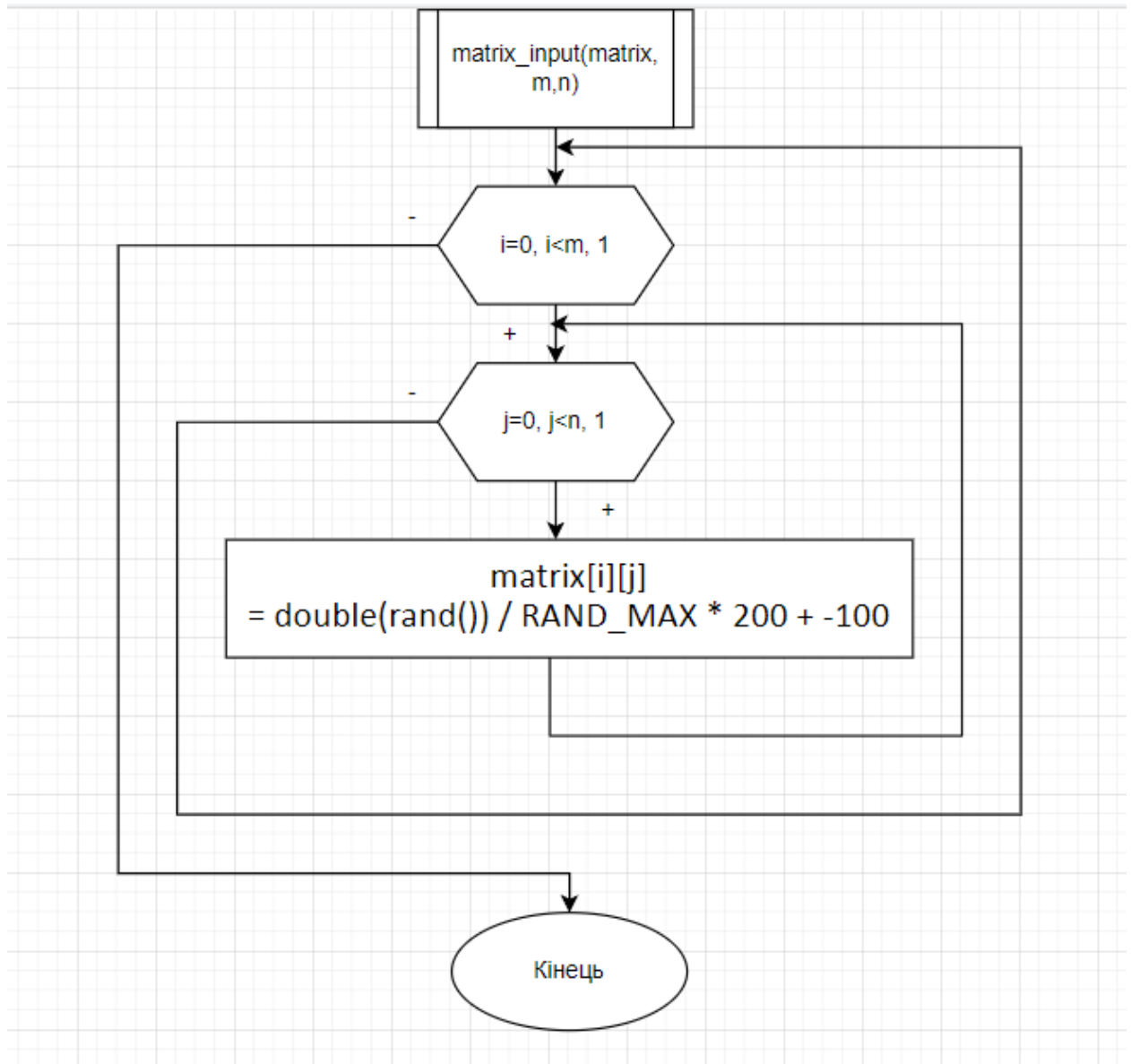
**Кінець**

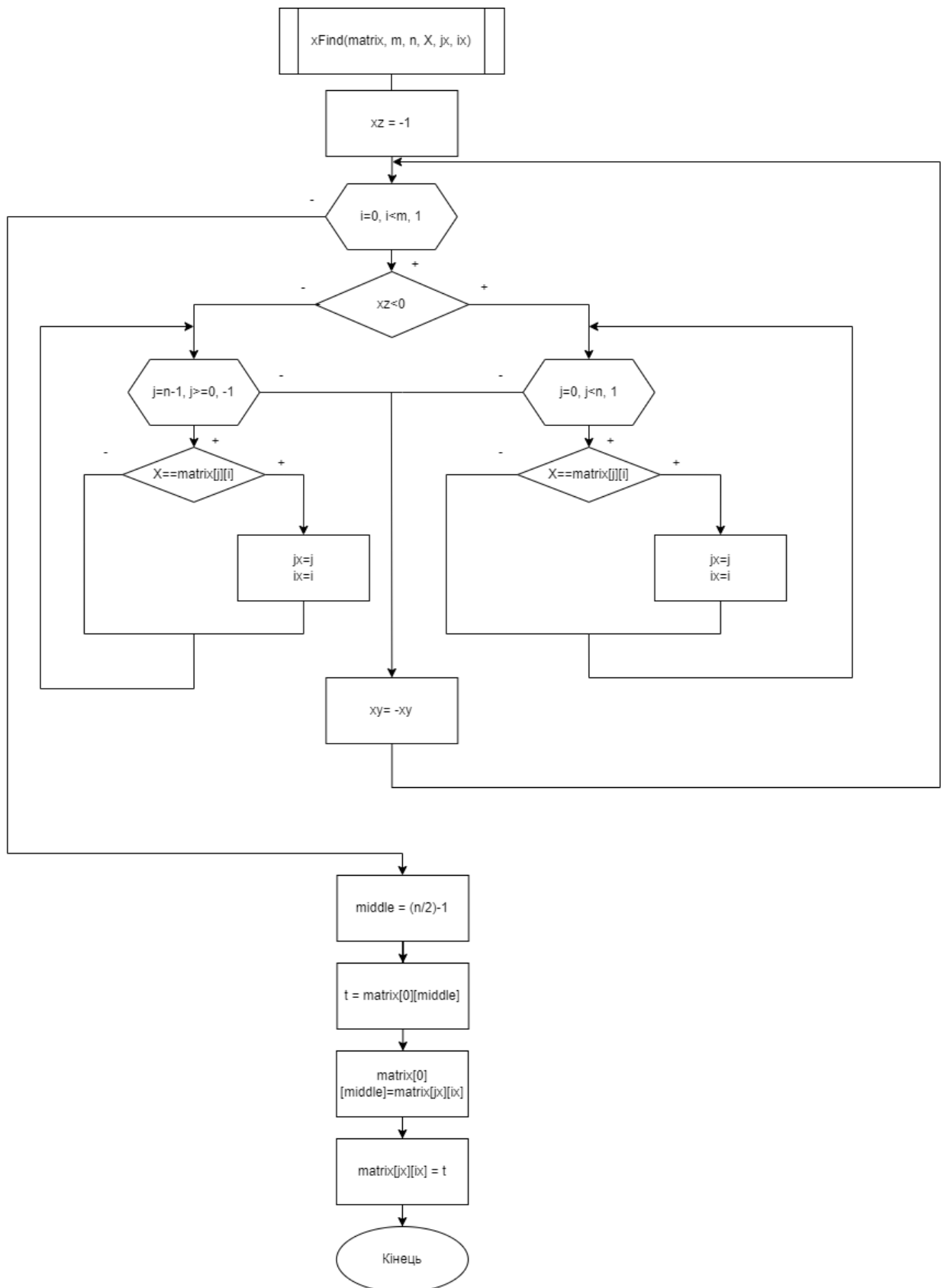
## Блок-схема

### Основна програма:



## Підпрограми:







## Код програми:

```
1  #include <iostream>
2  #include <ctime>
3  #include <cmath>
4  using namespace std;
5
6  void matrix_input(double matrix[10][10], int m, int n) {
7      cout << "Matrix:" << endl;
8      for (int i = 0; i < m; i++) {
9          for (int j = 0; j < n; j++) {
10             matrix[i][j] = double(rand()) / RAND_MAX * 200 + -100;
11             cout << matrix[i][j] << " ";
12         }
13         cout << endl;
14     }
15 }
16
17 void xFind(double matrix[10][10], int m, int n, double X, int jx, int ix) {
18     int xz = -1;
19     for (int i = 0; i < m; i++) {
20         if (xz < 0) {
21             for (int j = 0; j < n; j++) {
22                 if (int(X) == int(matrix[j][i])) {
23                     jx = j;
24                     ix = i;
25                     cout << "X found at: " << ix << "." << jx << endl;
26                 }
27             }
28         }
29         else {
30             for (int j = n - 1; j >= 0; j--) {
31                 if (int(X) == int(matrix[j][i])) {
32                     jx = j;
33                     ix = i;
34                     cout << "X found at: " << ix << "." << jx << endl;
35                 }
36             }
37         }
38         xz = -xz;
39     }
}
```

```

40     int middle = (int(n) / 2) - 1;
41     float t = matrix[0][middle];
42     matrix[0][middle] = matrix[jx][ix];
43     matrix[jx][ix] = t;
44 }
45
46
47
48 int main()
49 {
50     srand(time(NULL));
51     int m;
52     int n;
53     int jx = 1;
54     int ix = 1;
55     double x = 0;
56     cout << "Enter m, n:";
57     cin >> m >> n;
58     double matr[10][10];
59     matrix_input(matr, m, n);
60     cout << "Enter X:";
61     cin >> x;
62     xFind(matr, m, n, x, jx, ix);
63
64     cout << "changed matrix: " << endl;
65     for (int i = 0; i < m; i++) {
66         for (int j = 0; j < n; j++) {
67             cout << matr[i][j] << " ";
68         }
69         cout << endl;
70     }
71 }
72

```

## Випробування:

```

Enter m, n:5
6
Matrix:
-49.7665  12.4485  67.2231  -37.4493  45.9822  -57.6098
-51.8723  -59.1601  -51.5366  21.7322  53.2579  85.4732
66.1794  18.8391  -4.0376  -99.0234  -48.3566  -22.1961
38.9813  -85.8821  -20.2429  8.42616  49.5041  33.5856
-6.30207  -26.5603  -37.5225  -14.9388  -75.3044  54.2528
Enter X:21.7322
X found at: 3.1
changed matrix:
-49.7665  12.4485  21.7322  -37.4493  45.9822  -57.6098
-51.8723  -59.1601  -51.5366  67.2231  53.2579  85.4732
66.1794  18.8391  -4.0376  -99.0234  -48.3566  -22.1961
38.9813  -85.8821  -20.2429  8.42616  49.5041  33.5856
-6.30207  -26.5603  -37.5225  -14.9388  -75.3044  54.2528

```

**Висновки:**

Ми дослідили алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали програму для обходу заповненої матриці по стовпцям, пошуку заданого числа, виводу його розташування у матриці і заміни цього числа на елемент середнього стовпця.