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 лінійних алгоритмів ”
Варіант:12

Виконав студент: ІП-12 Єльчанінов Артем Юрійович
(шифр, прізвище, ім’я, по батькові)

Перевірив: _____
(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 12

Задача: Задано матрицю дійсних чисел A[m,n]. При обході матриці по стовпчиках знайти в ній останній мінімальний елемент і його місцезнаходження. Обміняти знайдене значення X з елементом першого рядка.

Постановка задачі

Результатом розв’язку задачі є матриця в якій останній мінімальний елемент поміняний місцем з елементом того ж стовпчика першого рядка.

Спершу вводимо бажану кількість рядків та стовпичків, потім для роботи з алгоритмом генеруються випадковим чином числа. Далі обходом матриці по стовпчиках знаходимо значення останнього мінімального елемента і його розташування. Останньою дією буде обмін значення останнього мінімального елемента (X) з елементом першого рядка. Після цього задача буде виконана.

Математична модель

Змінна	Тип	Ім’я	Призначення
Динамічна матриця	Дійсний	A	Вихідне дане
Кількість рядків матриці	Цілий	m	Вхідне дане
Кількість стовпців матриці	Цілий	n	Вхідне дане
Лічильник для арифметичних цилів	Цілий	i	Проміжне дане
Додатковий лічильник для арифметичного циклу	Цілий	j	Проміжне дане
Додатковий лічильник для арифметичного циклу	Цілий	k	Проміжне дане
Допоміжна змінна для обходу матриці	Цілий	temp_1	Проміжне дане
Функція для виведення матриці	Відсутній (void)	output_matrix	Допоміжний алгоритм
Змінна, що зберігає значення останнього	Дійсний	X	Проміжне дане

мінімального елемента матриці			
Змінна, що зберігає індекс рядка в якому розташований останній мінімальний елемент	Цілий	last_min_M	Проміжне дане
Змінна, що зберігає індекс стовпця в якому розташований останній мінімальний елемент	Цілий	last_min_N	Проміжне дане
Змінна, що зберігає значення елемента першого рядка з яким буде обміняний останній мінімальний елемент матриці	Дійсний	temp	Проміжне дане

Для генерації випадкових чисел будемо застосовувати функцію **rand( )**

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію заповнення матриці випадковими числами.

Крок 3. Деталізуємо дію визначення визначення значення останнього мінімального елемента матриці(X) та його розташування обходом матриці по стовпцям.

Крок 4. Деталізуємо дію обміну значення X з елементом першого рядка.

Псевдокод алгоритму

Крок 1:

Початок

Введення m, n

Перебір рядків матриці

Перебір стовпців матриці

Заповнення матрциі випадковими значеннями

Виведення значень матриці

Ініціалізація змінних

Обхід матриці по стовпцям

Знаходження X та його індексів

Обмін значення X з елементов першого рядка

Виведення значень зміненої матриці

Кінець

Крок 2:

Початок
  Введення m, n
    для i **від** 0 **до** m **з кроком** 1
      Перебір стовпців матриці
      Заповнення матрциі випадковими значеннями
    все повторити
    Виведення значень матриці
    Ініціалізація змінних
    Обхід матриці по стовпцям
    Знаходження X та його індексів
    Обмін значення X з елементов першого рядка
    Виведення значень зміненої матриці
Кінець

Крок 3:

Початок
  Введення m, n
    для i **від** 0 **до** m **з кроком** 1
      для j **від** 0 **до** n **з кроком** 1
        Заповнення матрциі випадковими значеннями
      все повторити
    все повторити
    Виведення значень матриці
    Ініціалізація змінних
    Обхід матриці по стовпцям
    Знаходження X та його індексів
    Обмін значення X з елементов першого рядка
    Виведення значень зміненої матриці
Кінець

Крок 4:

Початок
  Введення m, n
    для i **від** 0 **до** m **з кроком** 1
      для j **від** 0 **до** n **з кроком** 1
        A[i][j] := rand( ) % 100
      все повторити
    все повторити
    Виведення значень матриці
    Ініціалізація змінних
    Обхід матриці по стовпцям
    Знаходження X та його індексів
    Обмін значення X з елементов першого рядка
    Виведення значень зміненої матриці
Кінець

Крок 5:

Початок
  Введення m, n
    для i **від** 0 **до** m **з кроком** 1
      для j **від** 0 **до** n **з кроком** 1
        A[i][j] := rand( ) % 100
      все повторити
    все повторити
    output_matrix(A, m, n)
    Ініціалізація змінних
    Обхід матриці по стовпцям
    Знаходження X та його індексів
    Обмін значення X з елементов першого рядка
    Виведення значень зміненої матриці
Кінець

Крок 6:

Початок

Введення m, n

для i **від** 0 **до** m **з кроком** 1

для j **від** 0 **до** n **з кроком** 1

A[i][j] := rand( ) % 100

все повторити

все повторити

output_matrix(A, m, n)

X := A[0][0]; i := 0

Обхід матриці по стовпцям

Знаходження X та його індексів

Обмін значення X з елементов першого рядка

Виведення значень зміненої матриці

Кінець

Крок 7:

Початок
  Введення m, n
    для i **від** 0 **до** m **з кроком** 1
      для j **від** 0 **до** n **з кроком** 1
        A[i][j] := rand( ) % 100
      все повторити
    все повторити
    output_matrix(A, m, n)
    X := A[0][0]; i := 0
    для j **від** 0 **до** n **з кроком** 1
      якщо i == 0 | | i == -1
        то
          i := 0
          temp_1 := 1
        інакше якщо i == m
          то
            i := i – 1
            temp_1 := -1
      все якщо
      для k **від** 0 **до** m **з кроком** 1
        Знаходження X та його індексів
      все повторити
    все повторити
    Обмін значення X з елементов першого рядка
    Виведення значень зміненої матриці
Кінець

Крок 8:

Початок

Введення m, n

для i **від** 0 **до** m **з кроком** 1

для j **від** 0 **до** n **з кроком** 1

A[i][j] := rand( ) % 100

все повторити

все повторити

output_matrix(A, m, n)

X := A[0][0]; i := 0

для j **від** 0 **до** n **з кроком** 1

якщо i == 0 || i == -1

то

i := 0

temp_1 := 1

інакше якщо i == m

то

i := i – 1

temp_1 := -1

все якщо

для k **від** 0 **до** m **з кроком** 1

якщо X >= A[i][j]

то

X := A[i][j]

last_min_M := i

last_min_N := j

все якщо

i := i + temp_1

все повторити

все повторити

Обмін значення X з елементов першого рядка

Виведення значень зміненої матриці

Кінець

Крок 9:

Початок

Введення m, n

для i **від** 0 **до** m **з кроком** 1

для j **від** 0 **до** n **з кроком** 1

A[i][j] := rand( ) % 100

все повторити

все повторити

output_matrix(A, m, n)

X := A[0][0]; i := 0

для j **від** 0 **до** n **з кроком** 1

якщо i == 0 | | i == -1

то

i := 0

temp_1 := 1

інакше якщо i == m

то

i := i – 1

temp_1 := -1

все якщо

для k **від** 0 **до** m **з кроком** 1

якщо X >= A[i][j]

то

X := A[i][j]

last_min_M := i

last_min_N := j

все якщо

i := i + temp_1

все повторити

все повторити

temp := A[0][last_min_N]

A[0][last_min_N] := X

A[last_min_M][last_min_N] := temp

Виведення значень зміненої матриці

Кінець

Крок 10:

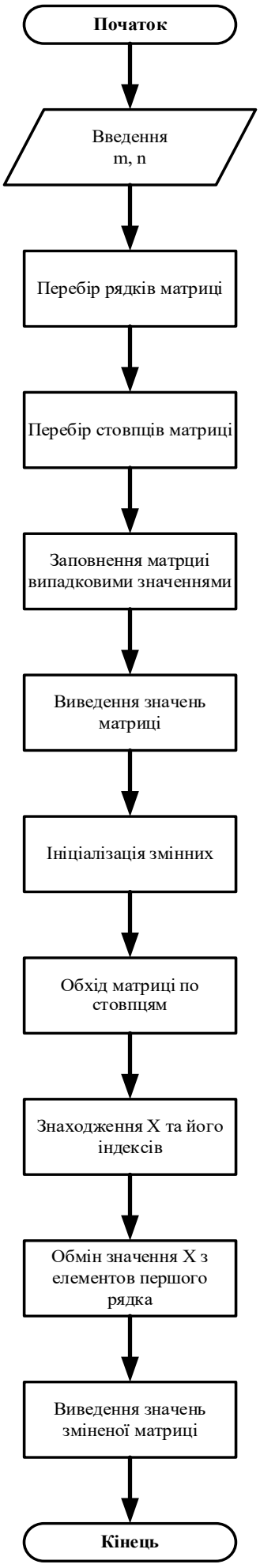
```
Початок
  Введення m, n
    для i від 0 до m з кроком 1
      для j від 0 до n з кроком 1
        A[i][j] := rand( ) % 100
      все повторити
    все повторити
    output_matrix(A, m, n)
    X := A[0][0];  i := 0
    для j від 0 до n з кроком 1
      якщо  i == 0  ||  i == -1
        то
          i := 0
          temp_1 := 1
        інакше якщо  i == m
          то
            i := i – 1
            temp_1 := -1
      все якщо
      для k від 0 до m з кроком 1
        якщо  X >= A[i][j]
          то
            X := A[i][j]
            last_min_M := i
            last_min_N := j
        все якщо
        i := i + temp_1
      все повторити
    все повторити
    temp := A[0][last_min_N]
    A[0][last_min_N] := X
    A[last_min_M][last_min_N] := temp
    output_matrix(A, m, n)
Кінець
```

output_matrix(A[m][n], m, n):

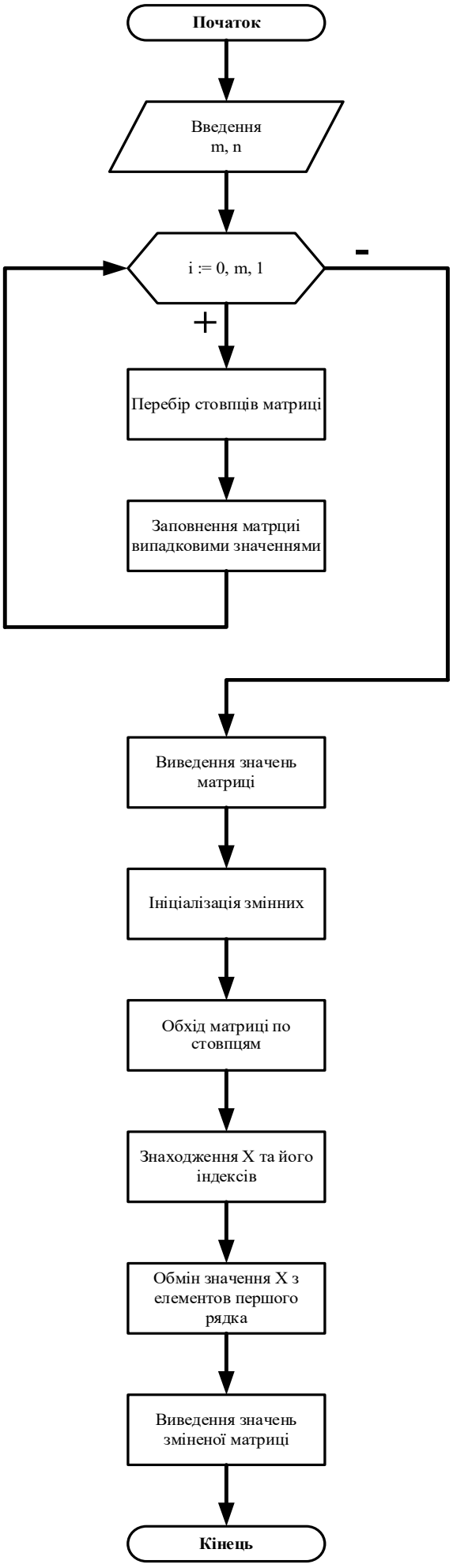
для i **від** 0 **до** m **з кроком** 1
для j **від** 0 **до** n **з кроком** 1
Виведення A[i][j] “\t”
все повторити
Виведення “\n”
все повторити
Кінець

Блок-схема

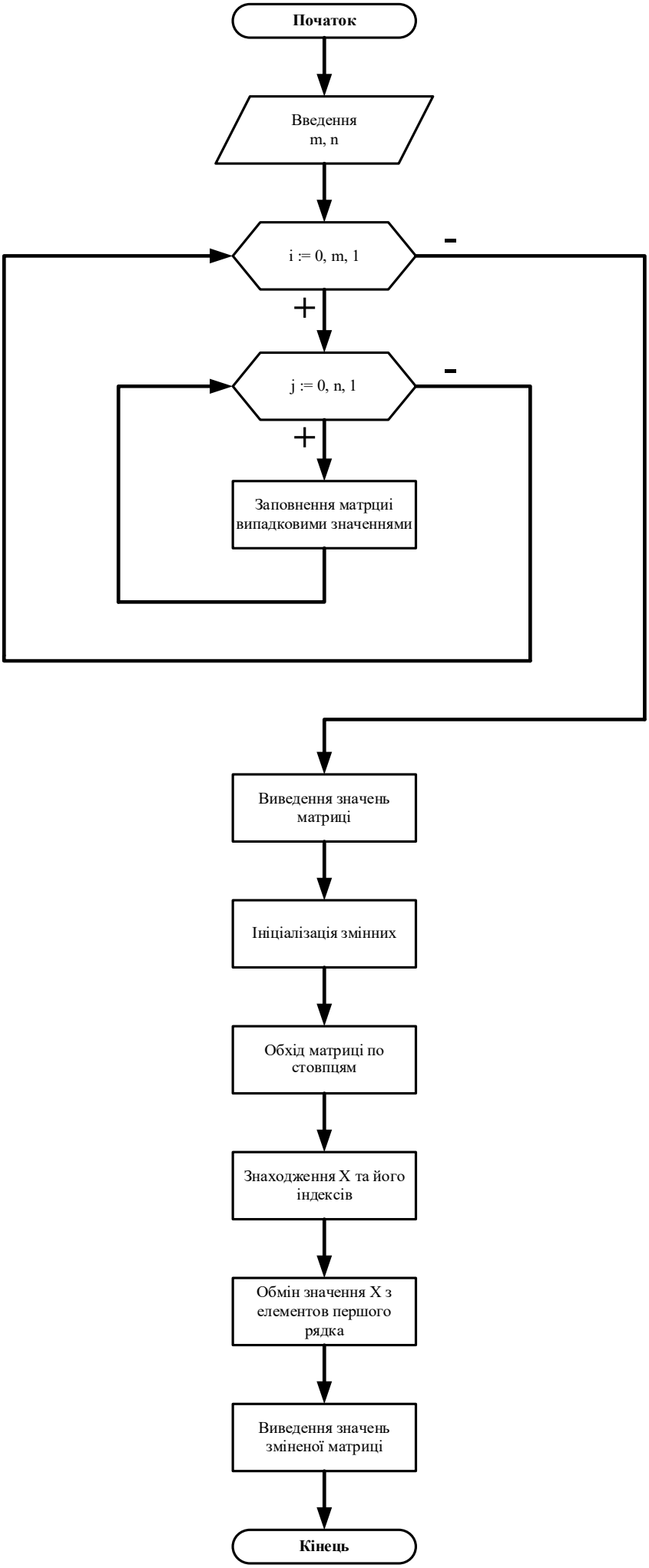
Крок 1:



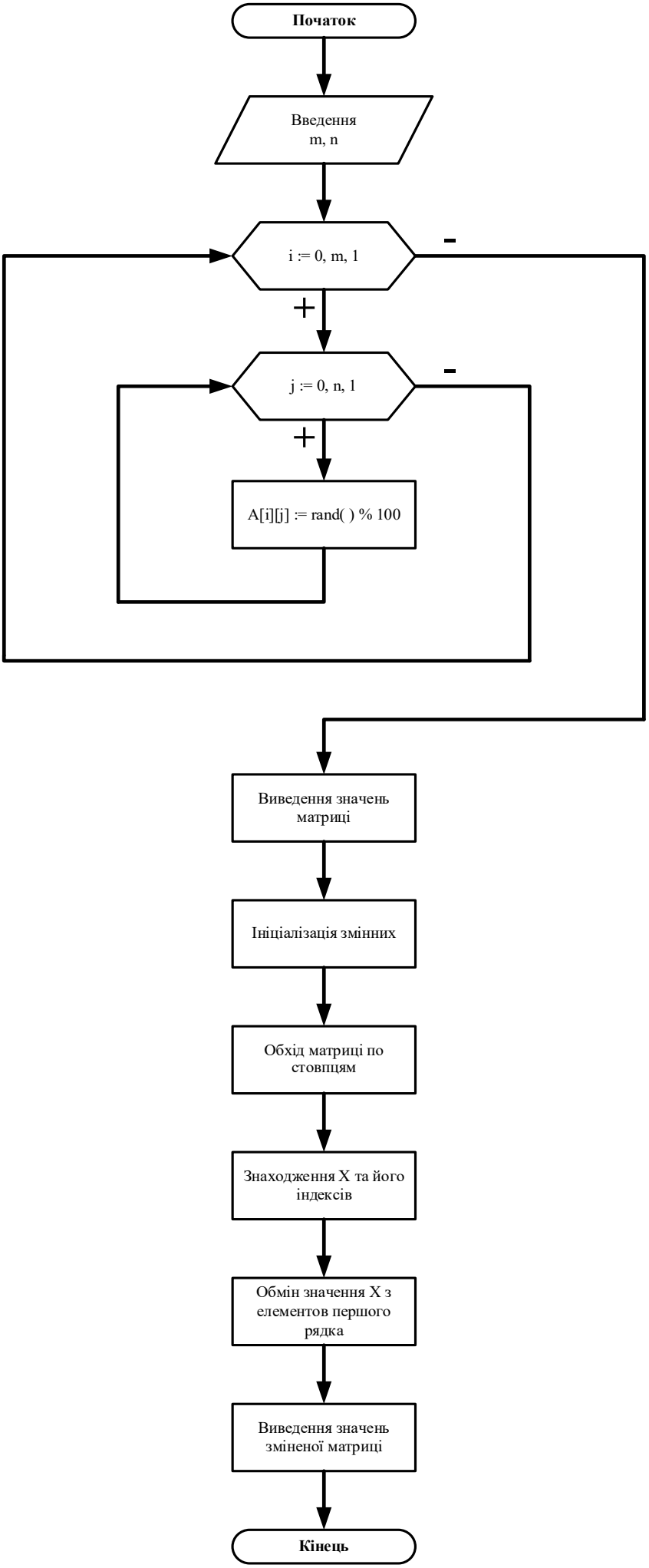
Крок 2:



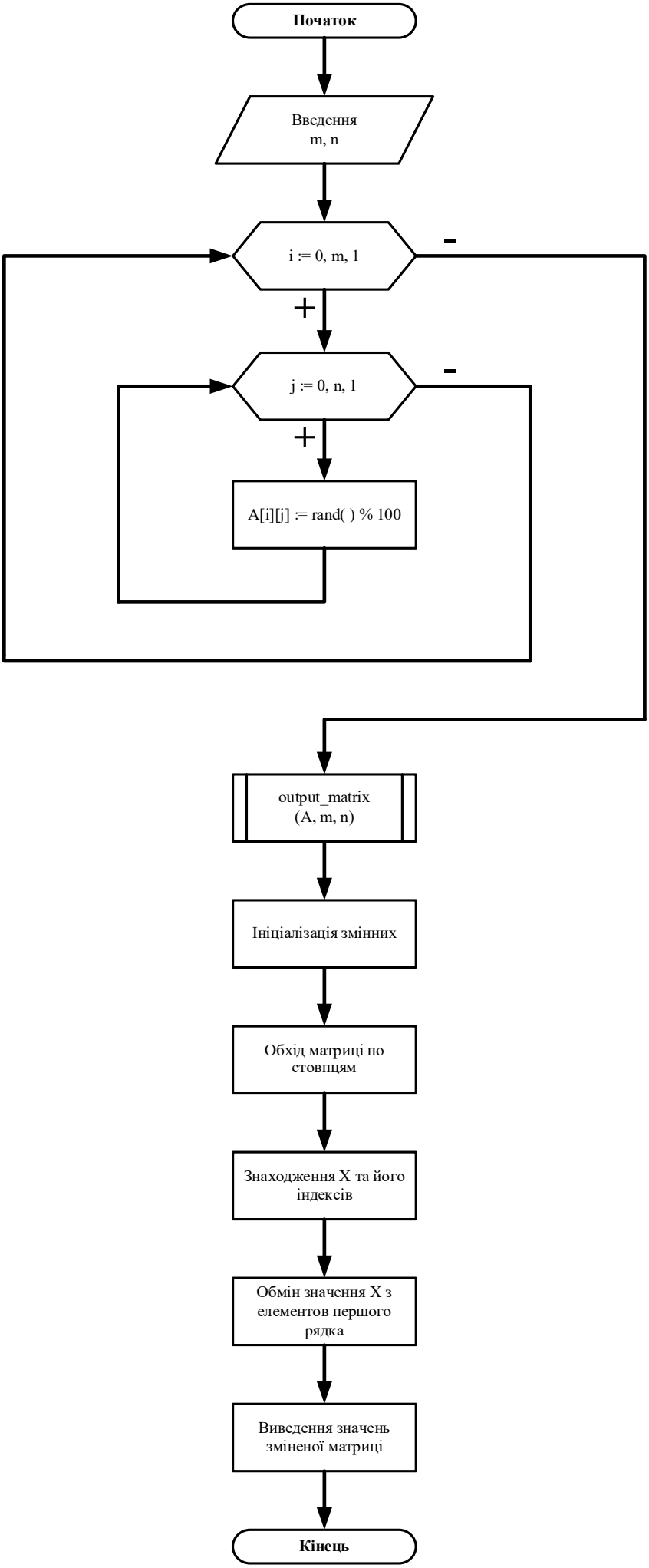
Крок 3:


Початок
Введення
m, n
i := 0, m, 1
-
+
j := 0, n, 1
-
+
Заповнення матрциі
випадковими значеннями
Виведення значень
матриці
Ініціалізація змінних
Обхід матриці по
стовпцям
Знаходження X та його
індексів
Обмін значення X з
елементов першого
рядка
Виведення значень
зміненої матриці
Кінець

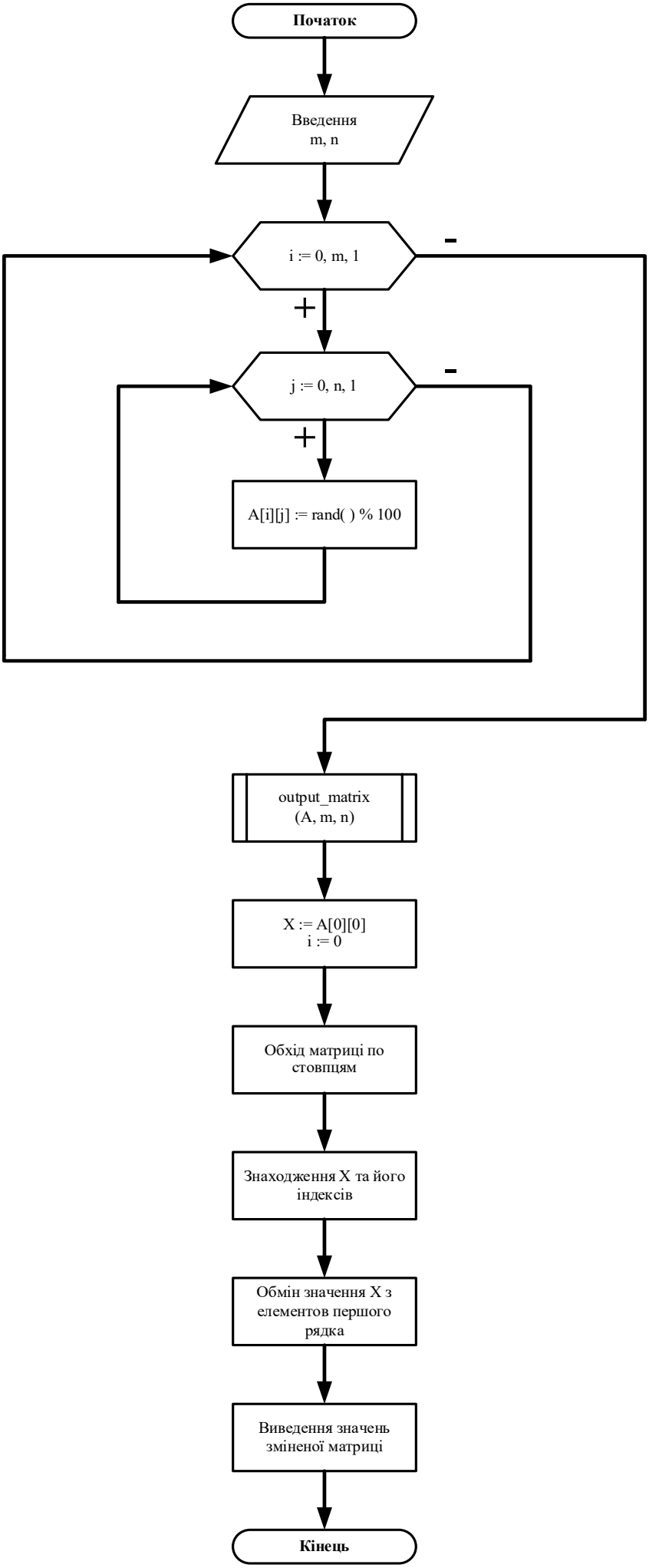
Крок 4:


Початок
Введення
m, n
i := 0, m, 1
-
+
j := 0, n, 1
-
+
A[i][j] := rand( ) % 100
Виведення значень
матриці
Ініціалізація змінних
Обхід матриці по
стовпцям
Знаходження X та його
індексів
Обмін значення X з
елементов першого
рядка
Виведення значень
зміненої матриці
Кінець

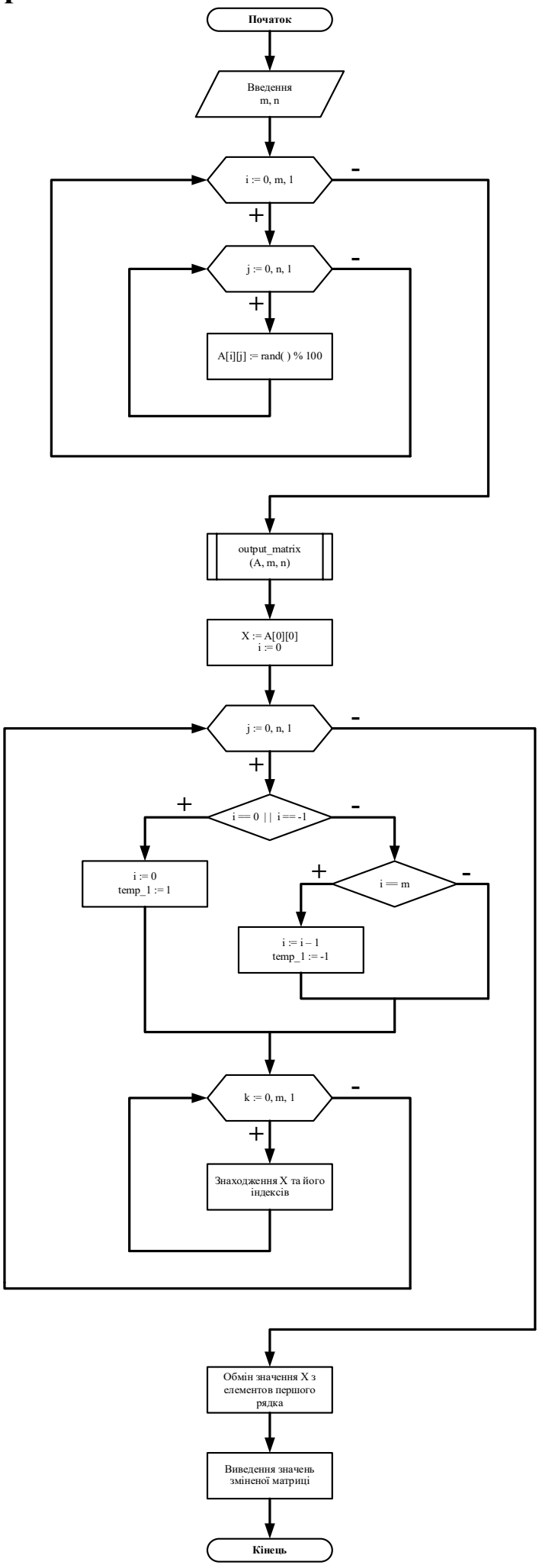
Крок 5:


Початок
Введення
m, n
i := 0, m, 1
-
+
j := 0, n, 1
-
+
A[i][j] := rand( ) % 100
output_matrix
(A, m, n)
Ініціалізація змінних
Обхід матриці по
стовпцям
Знаходження X та його
індексів
Обмін значення X з
елементов першого
рядка
Виведення значень
зміненої матриці
Кінець

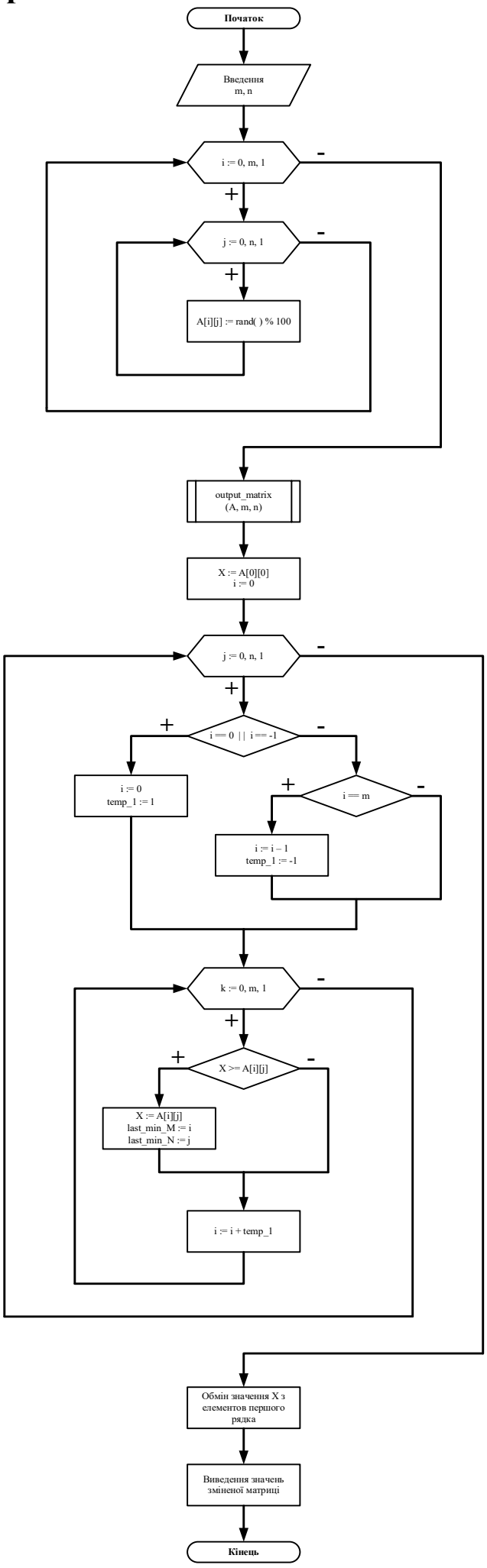
Крок 6:


Початок
Введення
m, n
i := 0, m, 1
-
+
j := 0, n, 1
-
+
A[i][j] := rand( ) % 100
output_matrix
(A, m, n)
X := A[0][0]
i := 0
Обхід матриці по стовпцям
Знаходження X та його індексів
Обмін значення X з елементов першого рядка
Виведення значень зміненої матриці
Кінець

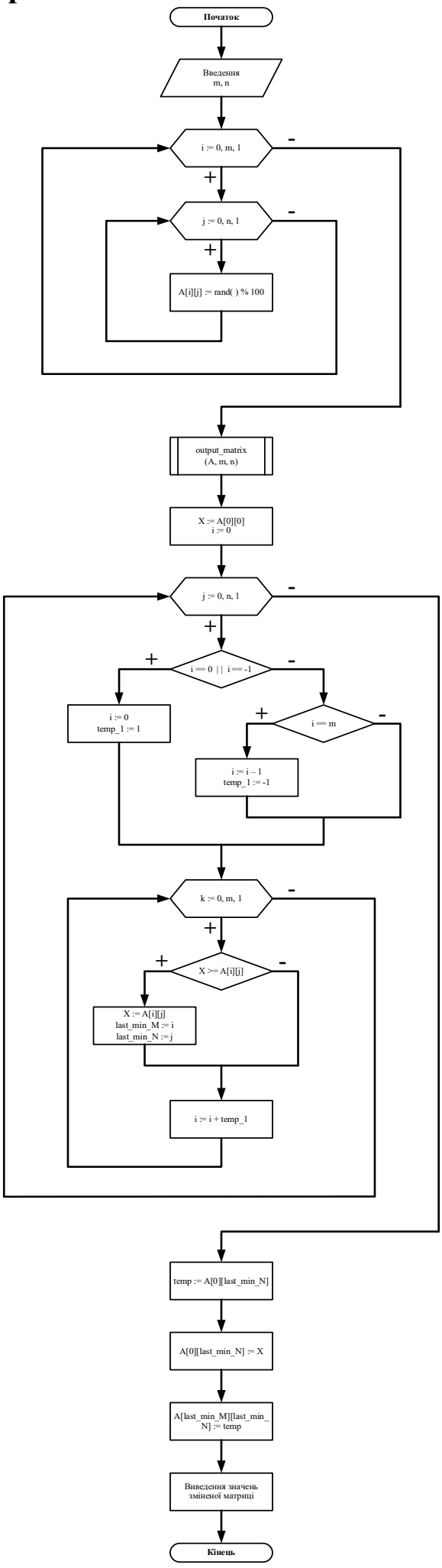
Крок 7:


Початок
Введення
m, n
i := 0, m, 1
-
+
j := 0, n, 1
-
+
A[i][j] := rand( ) % 100
output_matrix
(A, m, n)
X := A[0][0]
i := 0
j := 0, n, 1
-
+
+
i == 0 || i == -1
-
i := 0
temp_1 := 1
+
i == m
-
i := i – 1
temp_1 := -1
k := 0, m, 1
-
+
Знаходження X та його індексів
Обмін значення X з елементов першого рядка
Виведення значень зміненої матриці
Кінець

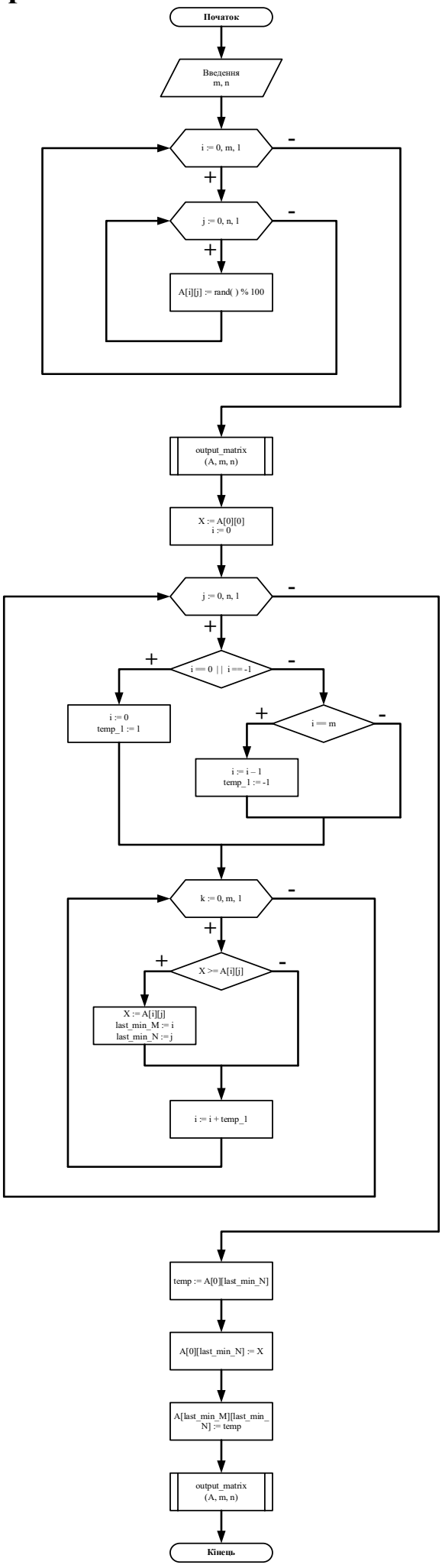
Крок 8:


Початок
Введення
m, n
i := 0, m, 1
j := 0, n, 1
A[i][j] := rand( ) % 100
output_matrix
(A, m, n)
X := A[0][0]
i := 0
j := 0, n, 1
i == 0 || i == -1
i := 0
temp_1 := 1
i == m
i := i – 1
temp_1 := -1
k := 0, m, 1
X >= A[i][j]
X := A[i][j]
last_min_M := i
last_min_N := j
i := i + temp_1
Обмін значення X з елементов першого рядка
Виведення значень зміненої матриці
Кінець

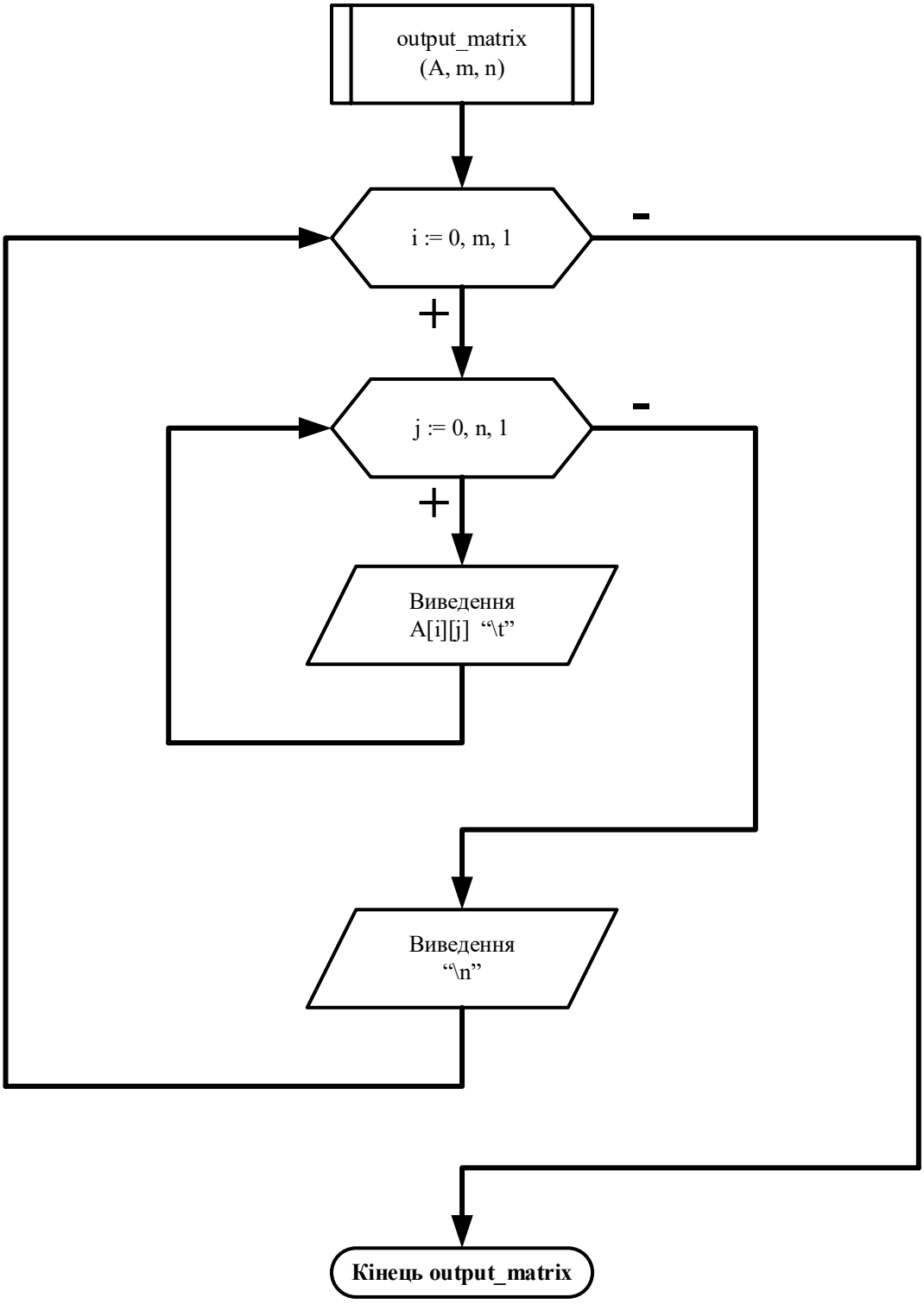
Крок 9:


Початок
Введення
m, n
i := 0, m, 1
j := 0, n, 1
A[i][j] := rand( ) % 100
output_matrix
(A, m, n)
X := A[0][0]
i := 0
j := 0, n, 1
i == 0 || i == -1
i := 0
temp_1 := 1
i == m
i := i – 1
temp_1 := -1
k := 0, m, 1
X >= A[i][j]
X := A[i][j]
last_min_M := i
last_min_N := j
i := i + temp_1
temp := A[0][last_min_N]
A[0][last_min_N] := X
A[last_min_M][last_min_N] := temp
Виведення значень зміненої матриці
Кінець

Крок 10:


Початок
Введення
m, n
i := 0, m, 1
j := 0, n, 1
A[i][j] := rand( ) % 100
output_matrix
(A, m, n)
X := A[0][0]
i := 0
j := 0, n, 1
i == 0 || i == -1
i := 0
temp_1 := 1
i == m
i := i – 1
temp_1 := -1
k := 0, m, 1
X >= A[i][j]
X := A[i][j]
last_min_M := i
last_min_N := j
i := i + temp_1
temp := A[0][last_min_N]
A[0][last_min_N] := X
A[last_min_M][last_min_N] := temp
output_matrix
(A, m, n)
Кінець

output_matrix(A[m][n], m, n):



Код програми на мові C++ :

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

void output_matrix(double**, int m, int n);

int main() {
    srand(time(NULL));
    int m, n;
    cout << "input number of lines: "; cin >> m;
    cout << "input number of columns: "; cin >> n;
    double** A = new double* [m];
    for (int i = 0; i < m; i++) {
        A[i] = new double[n];
    }
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            A[i][j] = rand() % 100;
        }
    }
    output_matrix(A, m, n);
    int i = 0;
    double X = A[0][0];
    int last_min_M, last_min_N, temp_1;
    for (int j = 0; j < n; j++) {
        if (i == 0 || i == -1) {
            i = 0;
            temp_1 = 1;
        }
        else if (i == m) {
            i = i - 1;
            temp_1 = -1;
        }
        for (int k = 0; k < m; k++) {
            if (X >= A[i][j]) {
                X = A[i][j];
                last_min_M = i;
                last_min_N = j;
            }
            i = i + temp_1;
        }
    }

    cout << "last_min: " << X << endl;
    cout << "line: " << last_min_M + 1 << "  ";
    cout << "column: " << last_min_N + 1 << "\n\n";

    double temp;
    temp = A[0][last_min_N];
    A[0][last_min_N] = X;
    A[last_min_M][last_min_N] = temp;
    output_matrix(A, m, n);

    for (int i = 0; i < m; i++) {
        delete[] A[i];
    }
    delete[] A;
    system("pause");
    return 0;
```

```
}

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

Тестування програми:

```
input number of lines: 4
input number of columns: 5
matrix:
36      79      69      47      2
6       10      17      46      15
42      66      47      56      12
6       25      96      83      80

last_min: 2
line: 1  column: 5

matrix:
36      79      69      47      2
6       10      17      46      15
42      66      47      56      12
6       25      96      83      80
```

```
input number of lines: 2
input number of columns: 10
matrix:
59      95      50      51      37      37      30      29      94      81
99      27      40      43      80      73      98      85      16      94

last_min: 16
line: 2  column: 9

matrix:
59      95      50      51      37      37      30      29      16      81
99      27      40      43      80      73      98      85      94      94
```

```
input number of lines: 8
input number of columns: 8
matrix:
64      29      61      86      27      8       45      40
23      36      58      12      85      21      88      68
2       16      92      77      3       25      6       17
43      78      98      70      63      73      26      28
81      53      71      65      97      19      59      96
96      78      90      48      1       48      36      75
82      71      77      99      23      48      18      59
17      41      51      65      45      1       33      79

last_min: 1
line: 8  column: 6

matrix:
64      29      61      86      27      1       45      40
23      36      58      12      85      21      88      68
2       16      92      77      3       25      6       17
43      78      98      70      63      73      26      28
81      53      71      65      97      19      59      96
96      78      90      48      1       48      36      75
82      71      77      99      23      48      18      59
17      41      51      65      45      8       33      79
```

Висновок.

У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму. Було набуто практичного новичок у використанні алгоритмів обходу масивів та їх інтерпретації у блок-схеми і псевдокод.

Алгоритм був випробуваний 3 рази. І в кожному з них була сгенерована матриця, потім було знайдено останній мінімальний елемент(X) та його розташування. У підсумку, було отримано матрицю, в якій були обміняні останній мінімальний елемент(X) з елементом першого рядка. Таким чином, було доведено вірність складеного алгоритму. Отже, його можна застосовувати для обходу матриці по стовпцям, та знаходження останнього мінімального елемента, та обміну його значення з елементом першого рядка.