

Основи програмування – 1. Алгоритми та структури даних

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 флгоритмів обходу масивів»

Варіант 10

Виконав студент

ІП-15,Закірова Олександра Володимирівна  
(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.

#### Завдання №10

Задано матрицю дійсних чисел  $A[m,n]$ , ініціалізувати матрицю обходом по стовбцях. У кожному стовпчику матриці знайти останній максимальний елемент і його місцезнаходження. Обміняти знайдене значення  $X$  з елементом першого стовбця

**Постановка задачі.** Результатом виконання завдання є двовимірний масив. Для розв'язання потрібна лише розмірність двовимірного масиву. Інших початкових даних для розв'язку не потрібно.

**Математична побудова.** Складемо таблицю змінних.

Змінна	Тип	Ім'я	Призначення
Число $n$	Цілий	$n$	Проміжна змінна
Число $m$	Цілий	$m$	Проміжна змінна
Число $i$	Цілий	$i$	Проміжна змінна
Число $j$	Цілий	$j$	Проміжна змінна
Число $Matrix[n][m]$	Дійсний	Matrix	Двовимірний масив
$k$	Дійсний	$k$	Проміжна змінна
$max$	Дійсний	$max$	Проміжна змінна

<b>b</b>	цілий	b	Проміжна змінна
<b>a</b>	цілий	a	Проміжна змінна
<b>Результат</b>	Двовимірний масив	<b>Matrix[n][m]</b>	<b>Результат</b>

Таким чином, математичне формулювання задачі зводиться до генерації елементів двовимірного масиву, перебору цього масиву у пошуку найбільших елементів кожного стовпчика та заміни елементів першого стовпчику на знайдені значення.

### ***Розв'язання:***

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

**Крок 1.** Визначимо основні дії.

**Крок 2.** Деталізуємо підпрограми, а саме:

- 1) Деталізуємо дію ініціалізації двовимірного масиву змійкою по стовпцях.
- 2) Деталізуємо функцію пошуку максимального значення кожного стовпчика, заміни елементів першого стовпчику на знайдені значення, вивід цих елементів із індексами, та модифікованного масиву.

### ***Псевдокод***

*крок 1*

***початок***

Ініціалізація матриці дійсними числами змійкою по стовпцях

Перебір стовпців множини у пошуку максимальних значень кожного стовпчика

Вивід иаксимальних значень стовпчиків із індексами

Заміна відповідних елементів першого стовпцю на знайдені максимальні значення

***кінець***

*крок 2*

**початок**

Ввід n, m

double Matrix[n][m]

SnakeCols(Matrix, n, m)

MaxWithIndex(Matrix, n, m)

Вивід Matrix[n][m]

**кінець**

*Підпрограми*

**початок SnakeCols**

int i, j

double k = 0,5

для int j = 0, j < m, j++

    для int i = 0, i < n, i++

        якщо j%2 == 0

            Matrix[i][j] = (j + 1) \* n - i - 0.5

        иначе

            Matrix[i][j] = k

        k++

**все повторити**

**все повторити**

**кінець SnakeCols**

**початок MaxWithIndex**

**для** `int j = 0, j < m, j++`

`int max = 0`

**для** `int i = 0, i < n, i++`

**якщо** `Matrix[max][j] == Matrix[i][j]`

`max = i`

**все повторити**

**для** `int a = 0, a < n, a++`

**для** `int b = 0, b < m, b++`

**якщо** `Matrix[a][b] == Matrix[max][j]`

Вивід `Matrix[max][j]` in `[a + 1][b + 1]` position in matrix

`Matrix[a][0] = Matrix[max][j]`

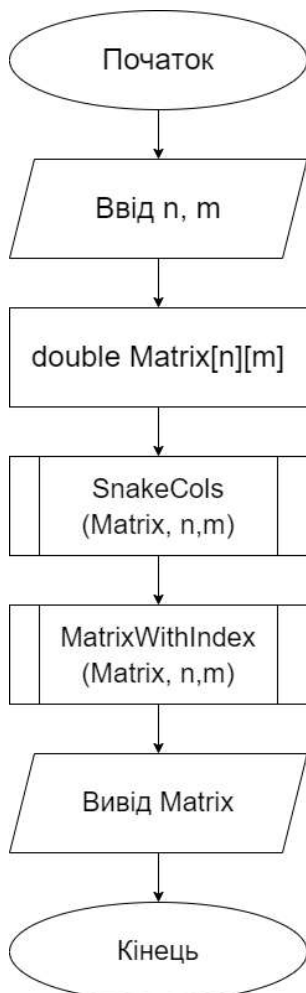
**все повторити**

**все повторити**

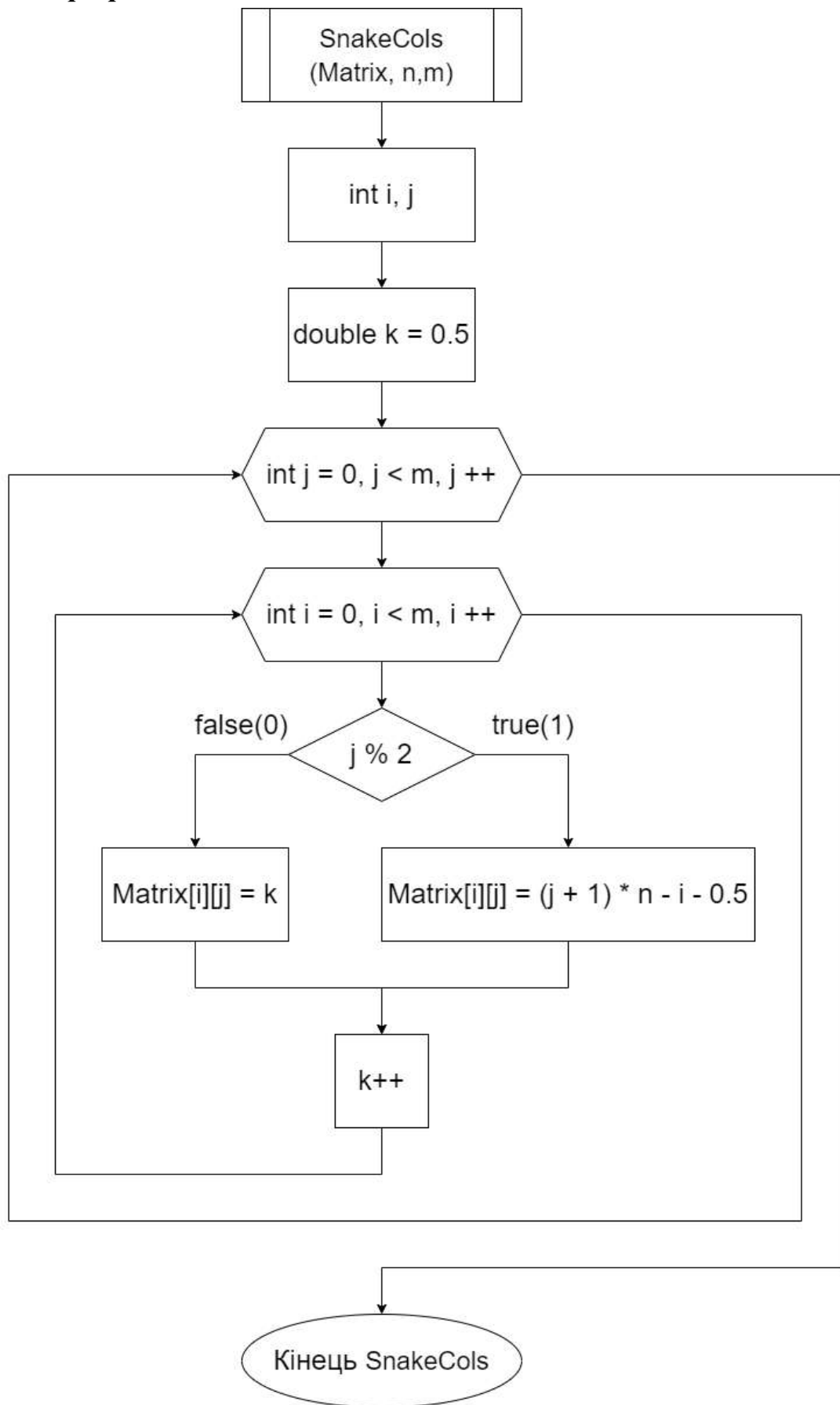
**все повторити**

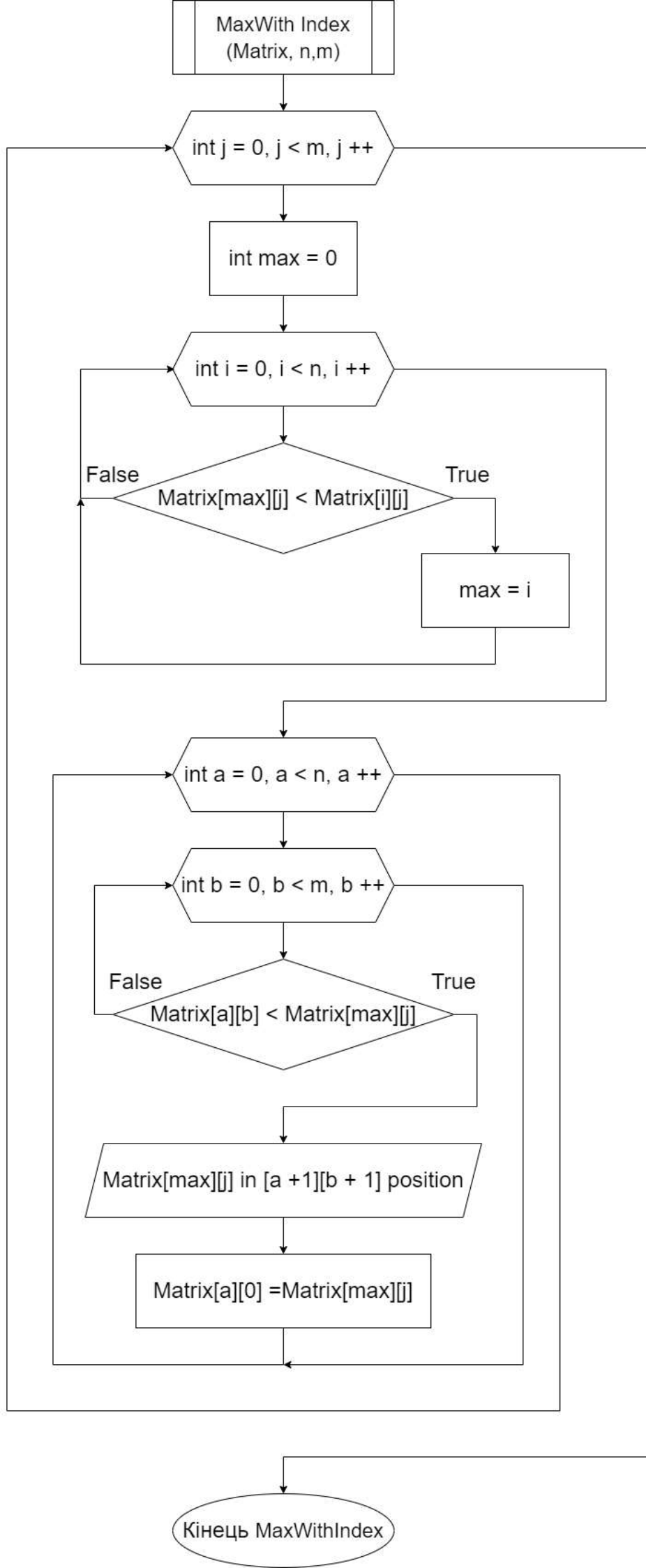
**кінець MaxWithIndex**

**Блок-схема**



## Підпрограми





## Код програми

```
CMakeLists.txt x main.cpp x
1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5
6  double **Create(size_t n, size_t m) {...}
13
14  void Delete(double **Matrix, int n) {...}
20
21  void SnakeCols(double **Matrix, int n, int m) {...}
31
32  void Print(double **Matrix, int n, int m) {...}
42
43  void MaxWithIndex(double **Matrix, int n, int m) {...}
62
63
64  int main() {
65      int n, m;
66      cout << "Set the dimension of the matrix A(n x m)\n";
67      cout << "n: ";
68      cin >> n;
69      cout << endl << "m: ";
70      cin >> m;
71      double **Matrix = Create(n, m);
72      SnakeCols(Matrix, n, m);
73      Print(Matrix, n, m);
74      MaxWithIndex(Matrix, n, m);
75      Print(Matrix, n, m);
76      Delete(Matrix, n);
77      return 0;
78 }
```



## Підпрограми

```
CMakeLists.txt x main.cpp x
5
6 double **Create(size_t n, size_t m) {
7     auto Matrix = new double *[n];
8     for (size_t i = 0; i < n; ++i) {
9         Matrix[i] = new double[m];
10    }
11    return Matrix;
12 }
13
14 void Delete(double **Matrix, int n) {
15     for (int i = 0; i < n; ++i) {
16         delete[] Matrix[i];
17     }
18     delete[] Matrix;
19 }
20
21 void SnakeCols(double **Matrix, int n, int m) {
22     int i, j;
23     double k = 0.5;
24     for (j = 0; j < m; j++) {
25         for (i = 0; i < n; i++) {
26             Matrix[i][j] = (j % 2 ? (j + 1) * n - i - 0.5 : k);
27             k++;
28         }
29     }
30 }
31
32 void Print(double **Matrix, int n, int m) {
33     int i, j;
34     for (i = 0; i < n; i++) {
35         cout << "\n";
36         for (j = 0; j < m; j++) {
37             cout << setw(4) << Matrix[i][j] << " ";
38         }
39     }
40     cout << "\n" << endl;
41 }
42
43 void MaxWithIndex(double **Matrix, int n, int m) {
44     for (int j = 0; j < m; j++) {
45         int max = 0;
46         for (int i = 0; i < n; i++) {
47             if (Matrix[max][j] < Matrix[i][j]) {
48                 max = i;
49             }
50         }
51         for (int a = 0; a < n; a++) {
52             for (int b = 0; b < m; b++) {
53                 if (Matrix[a][b] == Matrix[max][j]) {
54                     cout << "Max value in " << b + 1 << " column is " << Matrix[max][j] << " in [" << a + 1 << "]" <<
55                         << b + 1 << "]" position\n";
56                     Matrix[a][0] = Matrix[max][j];
57                 }
58             }
59         }
60     }
61 }
```

## Випробування алгоритму

```
Set the dimension of the matrix A(n x m)
n:6
m:7

0.5  11.5  12.5  23.5  24.5  35.5  36.5
1.5  10.5  13.5  22.5  25.5  34.5  37.5
2.5  9.5   14.5  21.5  26.5  33.5  38.5
3.5  8.5   15.5  20.5  27.5  32.5  39.5
4.5  7.5   16.5  19.5  28.5  31.5  40.5
5.5  6.5   17.5  18.5  29.5  30.5  41.5

Max value in 1 column is 5.5 in [6][1] position
Max value in 2 column is 11.5 in [1][2] position
Max value in 3 column is 17.5 in [6][3] position
Max value in 4 column is 23.5 in [1][4] position
Max value in 5 column is 29.5 in [6][5] position
Max value in 6 column is 35.5 in [1][6] position
Max value in 7 column is 41.5 in [6][7] position

35.5  11.5  12.5  23.5  24.5  35.5  36.5
1.5   10.5  13.5  22.5  25.5  34.5  37.5
2.5   9.5   14.5  21.5  26.5  33.5  38.5
3.5   8.5   15.5  20.5  27.5  32.5  39.5
4.5   7.5   16.5  19.5  28.5  31.5  40.5
41.5  6.5   17.5  18.5  29.5  30.5  41.5

Process finished with exit code 0
```

```
Set the dimension of the matrix A(n x m)
n:3
m:5

0.5  5.5  6.5  11.5  12.5
1.5  4.5  7.5  10.5  13.5
2.5  3.5  8.5   9.5  14.5

Max value in 1 column is 2.5 in [3][1] position
Max value in 2 column is 5.5 in [1][2] position
Max value in 3 column is 8.5 in [3][3] position
Max value in 4 column is 11.5 in [1][4] position
Max value in 5 column is 14.5 in [3][5] position

11.5  5.5  6.5  11.5  12.5
1.5   4.5  7.5  10.5  13.5
14.5  3.5  8.5   9.5  14.5

Process finished with exit code 0
```

## Висновок

Під час виконання лабораторної були досліджені особливості роботи алгоритмів обходу масивів та набуто практичних навичок їх використання під час складання програмних специфікацій. Покращено навички написання псевдокоду, побудови та тестування алгоритмів.