

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 лінійного пошуку в послідовностях»
Варіант: 19
Виконав студент: ІІІ-15 Левченко Владислав В'ячеславович
(шифр, прізвище, ім'я, по батькові)
Перевірила: Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)
Київ 2021

19	Задано матрицю дійсних чисел $A[m,n]$. У кожному рядку матриці знайти перший нульовий елемент X і його місцезнаходження. Обміняти знайдене значення X з елементом останнього стовбця.
----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

Побудова математичної моделі

Для вирішення даної задачі будемо використовувати три функції. Функція **fillArray()**, яка наповнює утворений двовимірний масив випадковими значеннями. Функція **checkArray()**, яка міняє перший нульовий елемент рядку з елементом останнього стовбця. І функція **printArray()**, яка виводить масив після утворення і після виконання задачі.

Таблиця імен зманих

Основна програма:

Змінна	Тип	Ім'я	Призначення
Рядки	Натуральне	row	Вхідне дане
Стовпці	Натуральне	col	Вхідне дане
Масив	Дісний	r	Вхідне дане, вихідне дане
Показчик	Ціле	i	Проміжне дане

Підпрограма fillArray():

Змінна	Тип	Ім'я	Призначення
Показчик внутрішнього циклу	Ціле	j	Проміжне дане
Масив	Дісний	arg	Вхідне дане, вихідне дане
Показчик зовнішнього циклу	Ціле	i	Проміжне дане
Рядки	Натуральне	m	Вхідне дане, проміжне дане
Стовпці	Натуральне	n	Вхідне дане, проміжне дане

Підпрограма printArray():

Змінна	Тип	Ім'я	Призначення
Показчик внутрішнього циклу	Ціле	j	Проміжне дане
Масив	Дісний	arg	Вхідне дане, вихідне дане
Показчик зовнішнього циклу	Ціле	i	Проміжне дане
Рядки	Натуральне	m	Вхідне дане, проміжне дане
Стовпці	Натуральне	n	Вхідне дане, проміжне дане

Підпрограма checkArray():

Змінна	Тип	Ім'я	Призначення
Показчик внутрішнього циклу	Ціле	j	Проміжне дане
Масив	Дісний	arr	Вхідне дане, вихідне дане
Показчик зовнішнього циклу	Ціле	i	Проміжне дане
Рядки	Натуральне	m	Вхідне дане, проміжне дане
Стовпці	Натуральне	n	Вхідне дане, проміжне дане

Псевдокод:

Крок 1. Визначимо основні дії

Крок 2. Створення підпрограм

Крок 3. Створення основної програми

Крок 1.

початок

Створення основної програми

Створення підпрограм

кінець

Крок 2.

початок

double** fillArray(double**, int, int)

void printArray(double**, int, int)

double** checkArray(double**, int, int)

Основна програма:

row: = rand() % 4 + 3

col: = rand() % 4 + 3

виведення row і col

****p: = new double* [row]**

для і від 0 до row

p[i]: = new double[col]

p: = fillArray(p, row, col)

printArray(p, row, col)

p: = checkArray(p, row, col)

printArray(p, row, col)

для і від 0 до row

delete[]p[i]

Створення підпрограм

кінець

Крок 3.

початок

double fillArray(double**, int, int)**

void printArray(double, int, int)**

double checkArray(double**, int, int)**

Основна програма:

row: = rand() % 4 + 3

col: = rand() % 4 + 3

виведення row і col

****p: = new double* [row]**

для і від 0 до row

p[i]: = new double[col]

```
p: = fillArray(p, row, col)
printArray(p, row, col)
p: = checkArray(p, row, col)
printArray(p, row, col)
```

```
    для і від 0 до row
        delete[]p[i]
```

Підпрограми:

```
fillArray(double** arr, const int m, const int n)
```

```
    для і від 0 до m
```

```
        для і від 0 до n
```

```
            arr[i][j]: = (double(rand()%5 - 1))
```

все повторити

повернення arr

```
printArray(double** arr, const int m, const int n)
```

```
    для і від 0 до m
```

```
        для і від 0 до n
```

```
            виведення arr[i][j]
```

все повторити

```
checkArray(double** arr, const int m, const int n)
```

```
    для і від 0 до m
```

```
        для і від 0 до n
```

```
            якщо arr[i][j] == 0.0
```

```
                то arr[i][j] = arr[i][n-1]
```

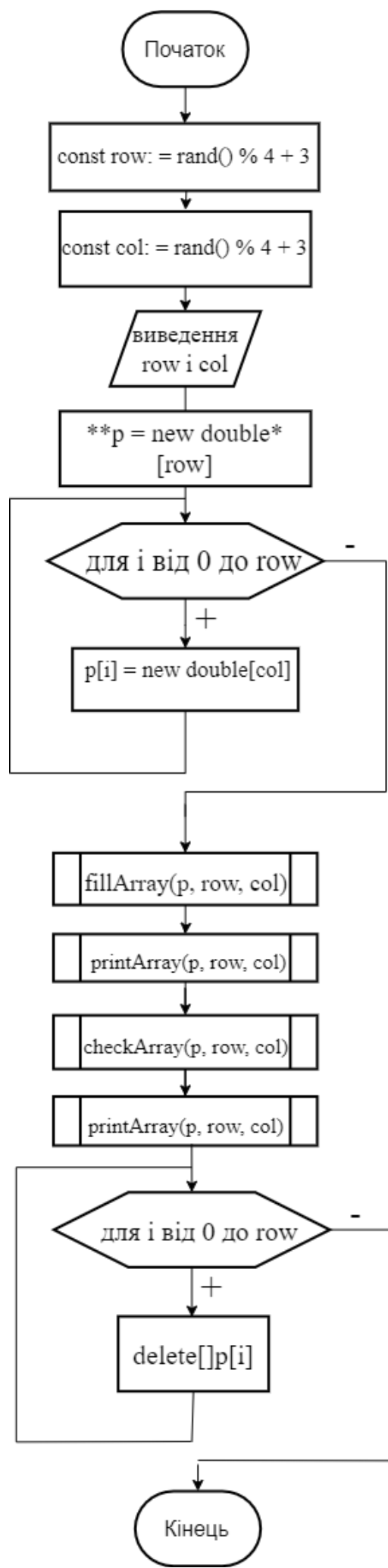
```
                arr[i][n-1] = 0
```

все якщо

все повторити
повернення arg
кінець

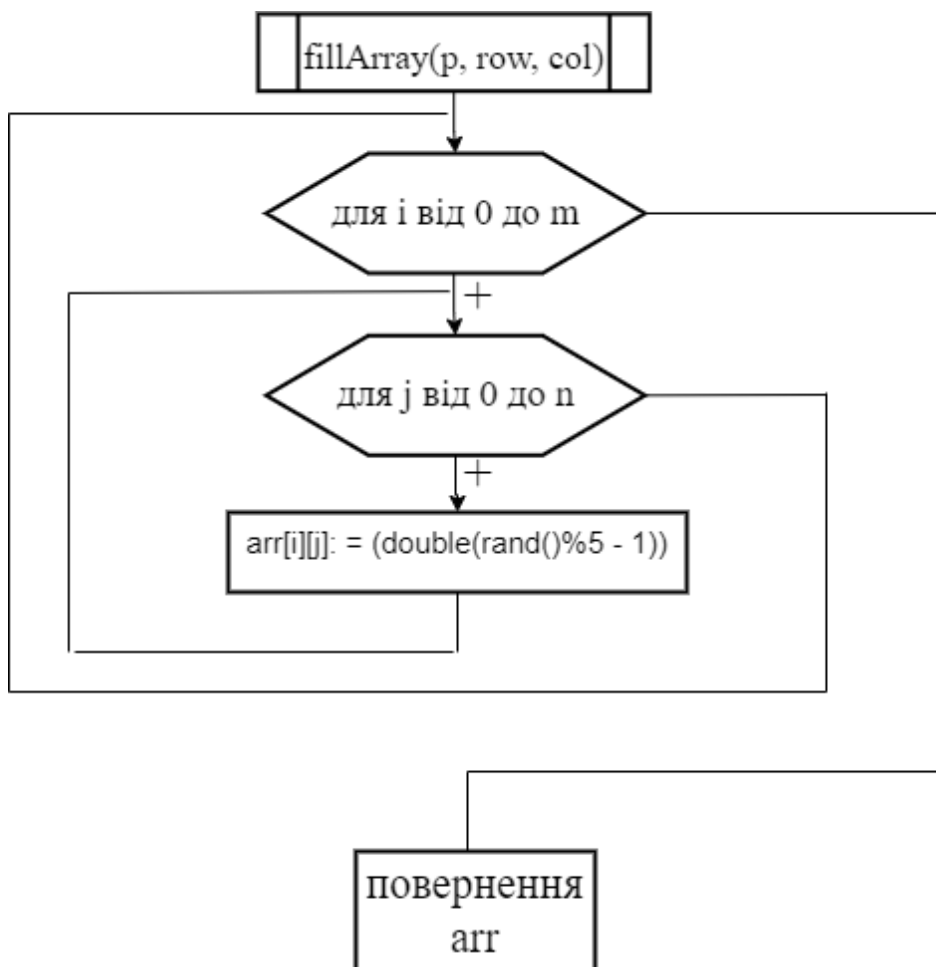
Блок схема

Основна програма:

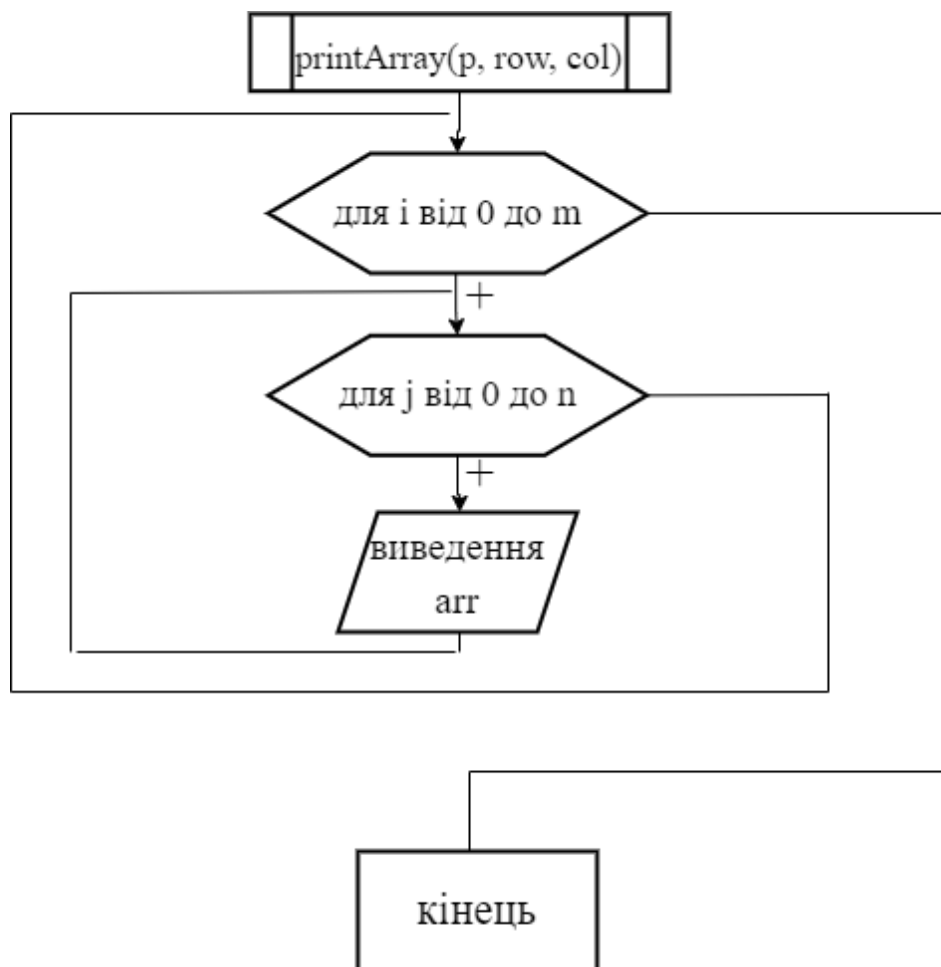


Підпрограми:

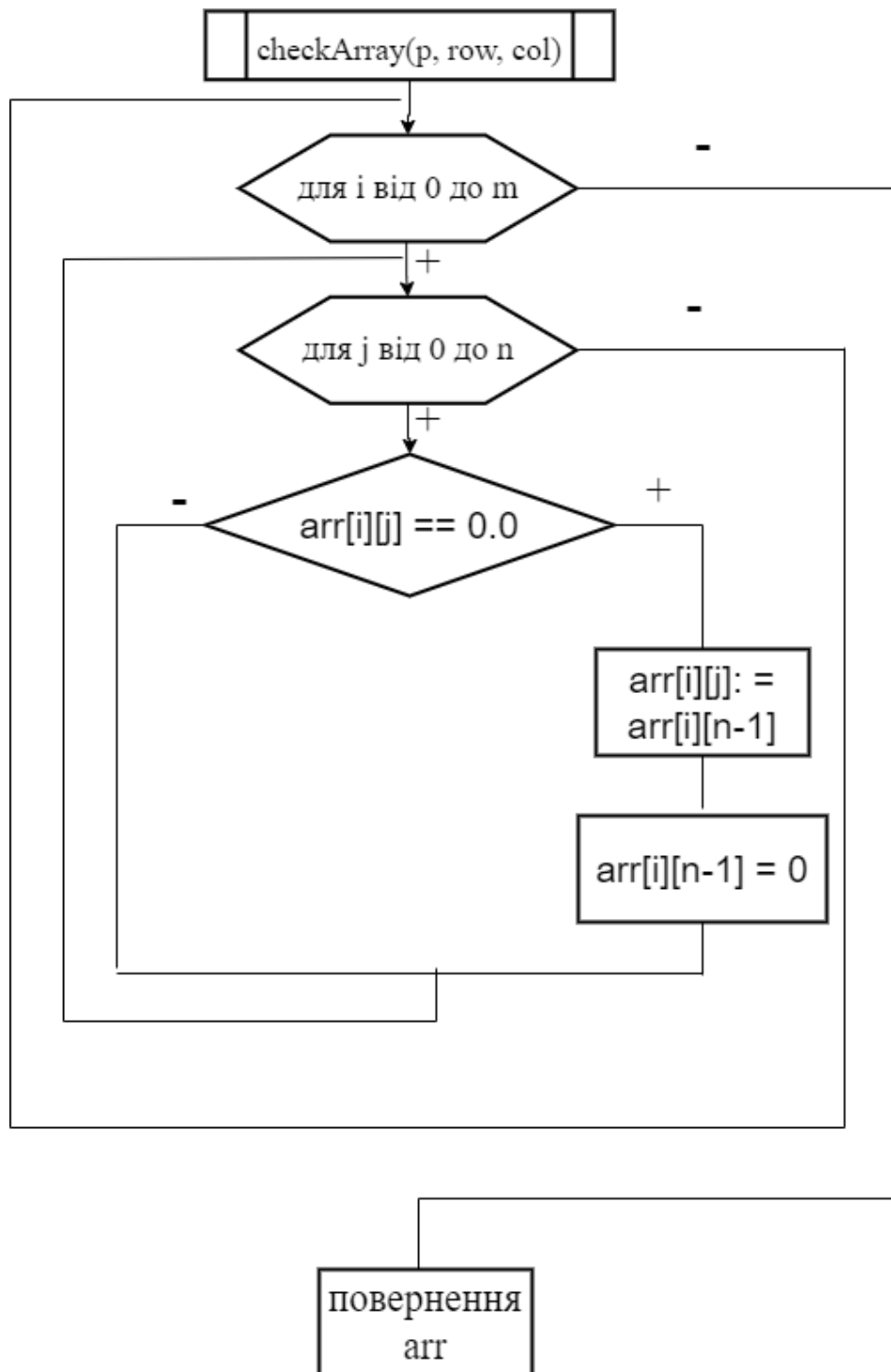
fillArray():



printArray():



`checkArray()`:



Код програми

```

1  #include <iostream>
2  #include <ctime>
3  #include <cstdio>
4  #include <cstdlib>
5  #include <iomanip>
6
7  using namespace std;
8
9  double** fillArray(double**, int, int);
10 void printArray(double**, int, int);
11 double** checkArray(double**, int, int);
12
13 int main()
14 {
15     srand(time(NULL));
16
17     const int row = rand() % 4 + 3;
18     const int col = rand() % 4 + 3;
19     cout << "rows: " << row << "\n" << "columns: " << col;
20
21     double** p = new double* [row];
22     for (int i = 0; i < row; i++) {
23         p[i] = new double[col];
24     }
25
26     p = fillArray(p, row, col);
27     printArray(p, row, col);
28     p = checkArray(p, row, col);
29     printArray(p, row, col);
30
31     for (int i = 0; i < row; i++) {
32         delete[]p[i];
33     }
34     delete[]p;
35     cin.get();
36 }
37
38 double** fillArray(double** arr, const int m, const int n) {
39     for (int i = 0; i < m; i++) {
40         for (int j = 0; j < n; j++) {
41             arr[i][j] = (double)(rand()%5 - 1);
42         }
43     }
44     return arr;

```

```

43     return arr;
44 }
45
46 void printArray(double** arr, const int m, const int n) {
47     cout << endl;
48     for (int i = 0; i < m; i++) {
49         cout << endl;
50
51         for (int j = 0; j < n; j++) {
52             if (arr[i][j] >= 0) { cout << " "; }
53             cout << fixed << setprecision(1) << arr[i][j] << " ";
54         }
55     }
56
57 double** checkArray(double** arr, const int m, const int n) {
58
59     cout << endl<< endl;
60     for (int i = 0; i < m; i++) {
61         for (int j = 0; j < n; j++) {
62
63             if (arr[i][j] == 0.0) {
64                 arr[i][j] = arr[i][n-1];
65                 arr[i][n-1] = 0;
66             }
67         }
68     }
69     return arr;
70 }

```

Випробування:

rows: 5 columns: 4	rows: 3 columns: 6
-1.0 -1.0 2.0 -1.0	0.0 1.0 2.0 0.0 2.0 0.0
-1.0 3.0 2.0 1.0	-1.0 3.0 3.0 0.0 1.0 2.0
-1.0 0.0 2.0 -1.0	-1.0 1.0 1.0 0.0 3.0 3.0
1.0 2.0 3.0 -1.0	
2.0 -1.0 -1.0 3.0	
-1.0 -1.0 2.0 -1.0	0.0 1.0 2.0 0.0 2.0 0.0
-1.0 3.0 2.0 1.0	-1.0 3.0 3.0 2.0 1.0 0.0
-1.0 -1.0 2.0 0.0	-1.0 1.0 1.0 3.0 3.0 0.0
1.0 2.0 3.0 -1.0	
2.0 -1.0 -1.0 3.0	

Висновки:

Під час виконання даної лабораторної роботи ми набули навичок обходу матриці, навчилися досліджувати та оперувати матрицею обійшовши її по спіралі.