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 рекурсивних алгоритмів» Варіант 2

Виконав студент ІП-14, Бабіч Денис Володимирович (шифр, прізвище, ім'я, по батькові)

Перевірила Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 2021

**Мета:** дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Варіант 2**

**Задача**:

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. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом.



**Постановка задачі:**

У алгоритмі використовується двовимірний масив 6 х 5, який заповнюється випадковими значеннями (в моєму випадку взяв проміжок від -100 до 100). Використовуються цілочисельні константи ROWS та COLUMNS для зберігання кількості рядків та стовпчиків матриці відповідно. Використовуються 2 масиви arr[ROWS][COLUMNS] та arr\_result[ROWS]. Також використовується змінна sum для зберігання суми всіх чисел рядка і обчислення середнього арифметичного кожного рядка. Змінні i, k, j та l – це ітератори у арифметичних циклах. Використовувуються функції rand() та srand() для генерування випадкових чисел для заповнення матриці. У тілі кожного арифметичного циклу зміну, до якої наближається ітератор вважати не включною, у даному алгоритмі у циклах відбувається лише інкрементування.

**Побудова математичної моделі:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Матриця | Дійсний  Двовимірний масив | arr | Матриця для розрахунків |
| Масив | Дійсний | arr\_result | Масив значень, проміжне значення |
| Зміння для збереження числа | Дійсний | sum | Значення середнього арифметичного |
| Збергіає числове значення | Ціле число | ROWS | Кількість рядків матриці |
| Збергіає числове значення | Ціле число | COLUMNS | Кількість стовпчиків матриці |
| Збергіає числове значення | Дійсне число | temp | Використовується  при обміні значеннями |
| Ітератор у арифметичному циклі | Ціле число | i | Ітератор у циклі |
| Ітератор у арифметичному циклі | Ціле число | k | Ітератор у циклі |
| Ітератор у арифметичному циклі | Ціле число | j | Ітератор у циклі |
| Ітератор у арифметичному циклі | Ціле число | l | Ітератор у циклі |

**Розв’язання**

Крок 1: Визначити основні дії

Крок 2: Ініціалізація початкових даних

Крок 3: Розрахунок середніх арифметичних

Крок 4: Сортування обміном за спаданням

Крок 5: Виведення отриманого масиву

**Псевдокод**

|  |  |
| --- | --- |
| Крок 1 | **Початок**  Визначити основні дії  Ініціалізація початкових даних  Розрахунок середніх арифметичних  Сортування обміном за спаданням  Виведення отриманого масиву  **Кінець** |
| Крок 2 | **Початок**  ROWS := 6  COLUMNS := 5  arr[ROWS][COLUMNS]  array\_result[ROWS]  sum := 0  Розрахунок середніх арифметичних  Сортування обміном за спаданням  Виведення отриманого масиву  **Кінець** |
| Крок 3 | **Початок**  ROWS := 6  COLUMNS := 5  arr[ROWS][COLUMNS]  array\_result[ROWS]  sum := 0  **повторити**  **для i від 0 до ROWS**  **повторити**  **для j від 0 до COLUMNS**  arr[i][j] = (rand() % 201) - 100  sum := sum + arr[i][j]  **все повторити**  sum := sum / COLUMNS  array\_result[i] := sum  sum = 0  **все повторити**  Сортування обміном за спаданням  Виведення отриманого масиву  **Кінець** |
|  |  |
| Крок 4 | **Початок**  ROWS := 6  COLUMNS := 5  arr[ROWS][COLUMNS]  array\_result[ROWS]  sum := 0  **повторити**  **для i від 0 до ROWS**  **повторити**  **для j від 0 до COLUMNS**  arr[i][j] = (rand() % 201) - 100  sum := sum + arr[i][j]  **все повторити**  sum := sum / COLUMNS  array\_result[i] := sum  sum = 0  **все повторити**  **повторити**  **для k від 0 до ROWS**  **повторити**  **для j від 0 до ROWS – k - 1**  **якщо array\_result[j] < array\_result[j + 1]**  **то**  temp := array\_result[j + 1]  array\_result[j + 1] := array\_result[j]  array\_result[j] := temp  **все якщо**  **все повторити**  **все повторити**  Виведення отриманого масиву  **Кінець** |
|  |  |

|  |  |
| --- | --- |
| Крок 5 | **Початок**  ROWS := 6  COLUMNS := 5  arr[ROWS][COLUMNS]  array\_result[ROWS]  sum := 0  **повторити**  **для i від 0 до ROWS**  **повторити**  **для j від 0 до COLUMNS**  arr[i][j] = (rand() % 201) - 100  sum := sum + arr[i][j]  **все повторити**  sum := sum / COLUMNS  array\_result[i] := sum  sum = 0  **все повторити**  **повторити**  **для k від 0 до ROWS**  **повторити**  **для j від 0 до ROWS – k - 1**  **якщо array\_result[j] < array\_result[j + 1]**  **то**  temp := array\_result[j + 1]  array\_result[j + 1] := array\_result[j]  array\_result[j] := temp  **все якщо**  **все повторити**  **все повторити**  **повторити**  **для l від 0 до ROWS**  **Виведення:** array\_result[l]  **все повторити**  **Кінець** |

**Блок-схема алгоритму**

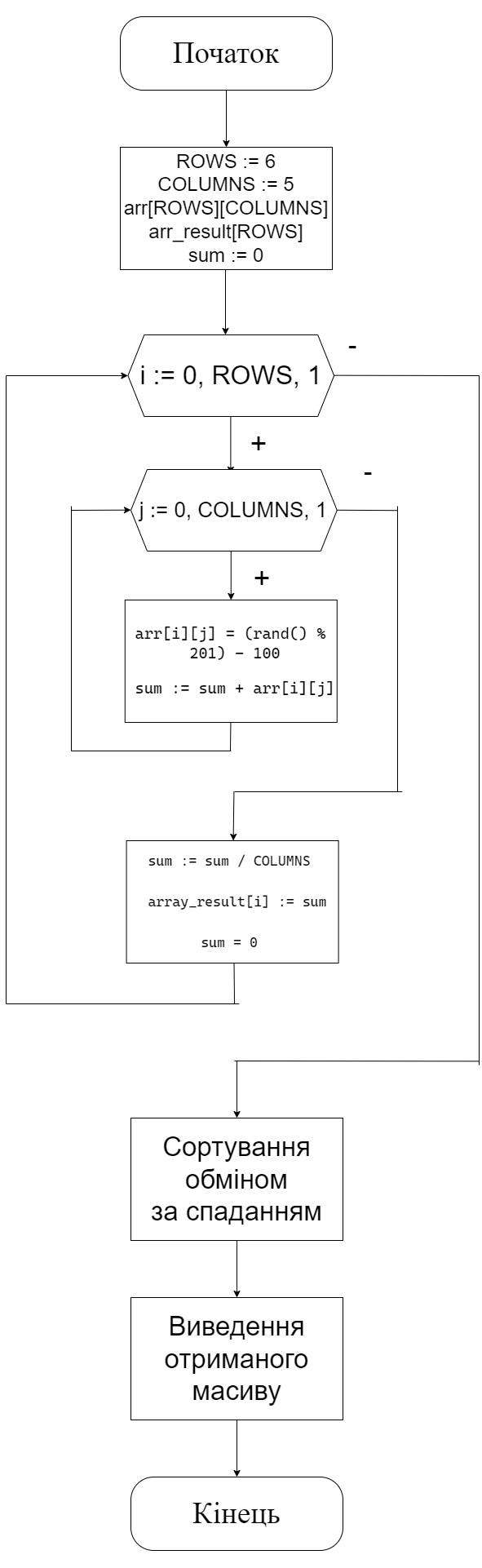
**Крок 1**

****

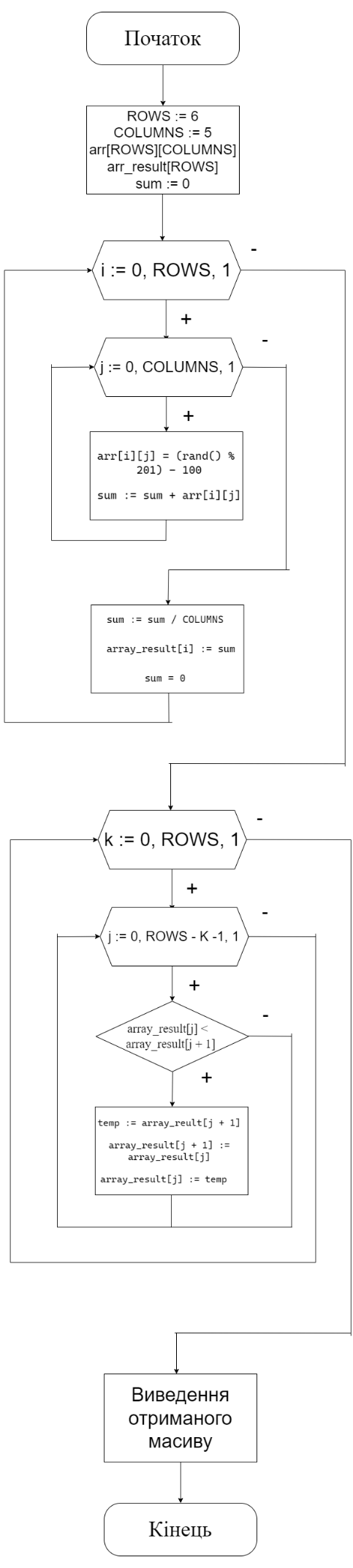
**Крок 2**

****

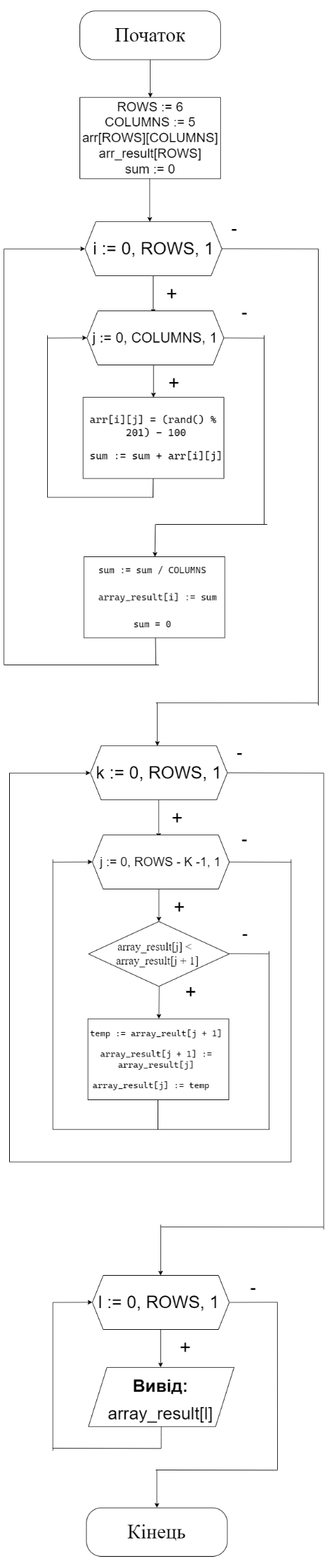
**Крок 3**

****

**Крок 4**

****

**Крок 5**

****

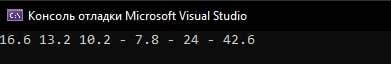
**Випробування алгоритму:**

**Код алгоритму**



**Випробування**

|  |  |
| --- | --- |
| Блок | Дія |
|  | **початок** |
| 1 | //Отримана матриця  0 48 -84 91 -94  48 -40 84 13 -22  -23 56 5 -93 -65  -43 -6 -30 -52 -82  69 -63 24 100 -64  54 67 -79 1 8  //Значення до сортування: 16.6 -7.8 -24 10.2 13.2 -42.6  //16.6 13.2 10.2 -7.8 -24 -42.6 – значення у масиві array\_result після сортування |
| 2 | Виведення: 16.6 13.2 10.2 -7.8 -24 -42.6 |
|  | **кінець** |
|  |  |



**Висновок:**

У результаті лабораторної роботи набув навичок у використані двовимірних масивів, під час виконання роботи використовував оператори альтернативного вибору, арифметичні операції, арифметичні цикли та функцію для створення випадкових чисел у заданому діапазоні. Дослідив алгоритм сортування обміном за спаданням.