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 алгоритмів пошуку та сортування»

Варіант 22

Виконав студент ІП-12 Мельник Михайло Олександрович

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 8**

**Дослідження алгоритмів пошуку та сортування**

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

**Варіант 22**

**Задача:**

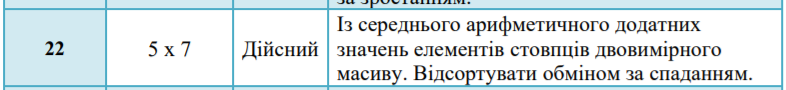
Розробити алгоритм та написати програму, яка складається з наступних

дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).

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 (табл. 1).



**Розв’язання:**

1. **Постановка задачі**

Результатом розв’язку задачі є відсортований обміном за спадання одновимірний масив, кожен елемент якого дорівнює середньому арифметичному додатних значень елементів стовпців двовимірного масиву. Для його визначення не потрібні початкові значення.

1. **Побудова математичної моделі**

***Складемо таблицю імен змінних***

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Двовимірний масив | Індексований дійсний | A | Результат |
| Одновимірний масив | Індексований дійсний | B | Результат |
| Діапазон значень елементів масиву | Цілий | RANGE | Початкове дане |
| Кількість рядків | Цілий | x | Початкове дане |
| Кількість стовпців | Цілий | y | Початкове дане |
| Лічильник i | Порядковий | i | Проміжне дане |
| Лічильник j | Порядковий | j | Проміжне дане |
| Сума додатних значень елементів стовпців двовимірного масиву | Дійсний | sum | Проміжне дане |
| Кількість додатних значень елементів стовпців двовимірного масиву | Цілий | n | Проміжне дане |
| Чи відбувся обмін | Логічний | changed | Проміжне дане |
| Тимчасова змінна для обміну | Дійсний | temp | Проміжне дане |

Таким чином, математичне формулювання задачі зводиться до:

* 1. Опису констант.
  2. Опису та ініціювання двовимірного масиву випадковими значеннями у проміжку від RANGE до -RANGE обходом за рядками.
  3. Опису та ініціювання одновимірного масиву обходом за стовпцями з лічильником j, у якому ми проходимо кожен елемент стовпця з лічильником i, додаємо значення додатних елементів до sum і збільшуємо n на 1, якщо n == 0, елемент індексу і набуває значення 0, інакше – sum / n.
  4. Сортування одновимірного масиву методом бульбашки, в якому ми обходимо усі елементи від 0 до y-1 поки changed==true, порівнюємо значення даного і наступного, і якщо наступний має більше значення, обмінюємо їх і надаємо changed значення true.
  5. Виводу двовимірного і одновимірного масиву обходом елементів.

1. **Розроблення алгоритму**

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію опису констант

*Крок 3.* Деталізуємо дію опсу та ініціювання двовимірного масиву.

*Крок 4.* Деталізуємо дію опису та ініціювання одновимірного масиву.

*Крок 5.* Деталізуємо дію сортування одновимірного масиву.

*Крок 6.* Деталізуємо дію виводу значень елементів масивів.

***Псевдокод***

*крок 1*

**початок**

опис констант

опис та ініціювання двовимірного масиву

опис та ініціювання одновимірного масиву

сортування одновимірного масиву

вивід значень елементів масивів.

**кінець**

*крок 2*

**початок**

x := 5

y := 7

RANGE := 100

опис та ініціювання двовимірного масиву

опис та ініціювання одновимірного масиву

сортування одновимірного масиву

вивід значень елементів масивів.

**кінець**

*крок 3*

**початок**

x := 5

y := 7

RANGE := 100

**повторити**

**для** і **від** 0 **до** х

**повторити**

**для** j **від** 0 **до** y

A[i][j] := rand() / RAND\_MAX \* 2 \* RANGE – RANGE

**все повторити**

**все повторити**

опис та ініціювання одновимірного масиву

сортування одновимірного масиву

вивід значень елементів масивів

**кінець**

*крок 4*

**початок**

x := 5

y := 7

RANGE := 100

**повторити**

**для** і **від** 0 **до** х

**повторити**

**для** j **від** 0 **до** y

A[i][j] := rand() / RAND\_MAX \* 2 \* RANGE – RANGE

**все повторити**

**все повторити**

sum := 0;

n := 0;

**повторити**

**для** j **від** 0 **до** y

**повторити**

**для** і **від** 0 **до** x

**якщо** A[i][j] > 0

**то**

sum := sum + A[i][j]

n := n + 1

**все якщо**

**все повторити**

**якщо** n != 0

**то**

B[j] := sum / n

**інакше**

B[j] := 0

**все якщо**

sum := 0

n := 0

**все повторити**

сортування одновимірного масиву

вивід значень елементів масивів

**кінець**

*крок 5*

**початок**

x := 5

y := 7

RANGE := 100

**повторити**

**для** і **від** 0 **до** х

**повторити**

**для** j **від** 0 **до** y

A[i][j] := rand() / RAND\_MAX \* 2 \* RANGE – RANGE

**все повторити**

**все повторити**

sum := 0;

n := 0;

**повторити**

**для** j **від** 0 **до** y

**повторити**

**для** і **від** 0 **до** x

**якщо** A[i][j] > 0

**то**

sum := sum + A[i][j]

n := n + 1

**все якщо**

**все повторити**

**якщо** n != 0

**то**

B[j] := sum / n

**інакше**

B[j] := 0

**все якщо**

sum := 0

n := 0

**все повторити**

changed := true

**повторити**

**поки** changed == true

changed := false

**повторити**

**для** і **від** 0 **до** y-1

**якщо** B[i] < B[i+1]

**то**

temp := B[i]

B[i] := B[i+1]

B[i+1] := temp

changed := true

**все якщо**

**все повторити**

**все повторити**

вивід значень елементів масивів

**кінець**

*крок 6*

**початок**

x := 5

y := 7

RANGE := 100

**повторити**

**для** і **від** 0 **до** х

**повторити**

**для** j **від** 0 **до** y

A[i][j] := rand() / RAND\_MAX \* 2 \* RANGE – RANGE

**все повторити**

**все повторити**

sum := 0;

n := 0;

**повторити**

**для** j **від** 0 **до** y

**повторити**

**для** і **від** 0 **до** x

**якщо** A[i][j] > 0

**то**

sum := sum + A[i][j]

n := n + 1

**все якщо**

**все повторити**

**якщо** n != 0

**то**

B[j] := sum / n

**інакше**

B[j] := 0

**все якщо**

sum := 0

n := 0

**все повторити**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

**.**

changed := true

**повторити**

**поки** changed == true

changed := false

**повторити**

**для** і **від** 0 **до** y-1

**якщо** B[i] < B[i+1]

**то**

temp := B[i]

B[i] := B[i+1]

B[i+1] := temp

changed := true

**все якщо**

**все повторити**

**все повторити**

**повторити**

**для** і **від** 0 **до** х

**повторити**

**для** j **від** 0 **до** y

**вивід** A[i][j]

**все повторити**

**все повторити**

**повторити**

**для** і **від** 0 **до** y

**вивід** B[i]

**все повторити**

**кінець**

***Блок-схема***

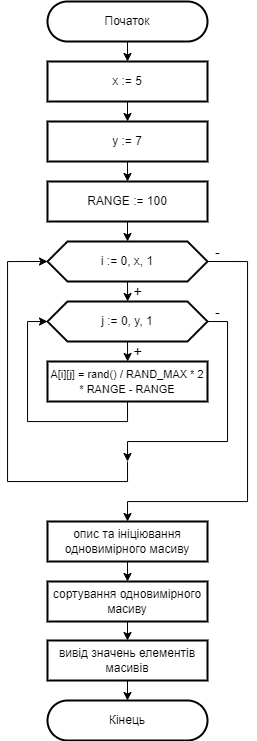
*Крок 1*

**

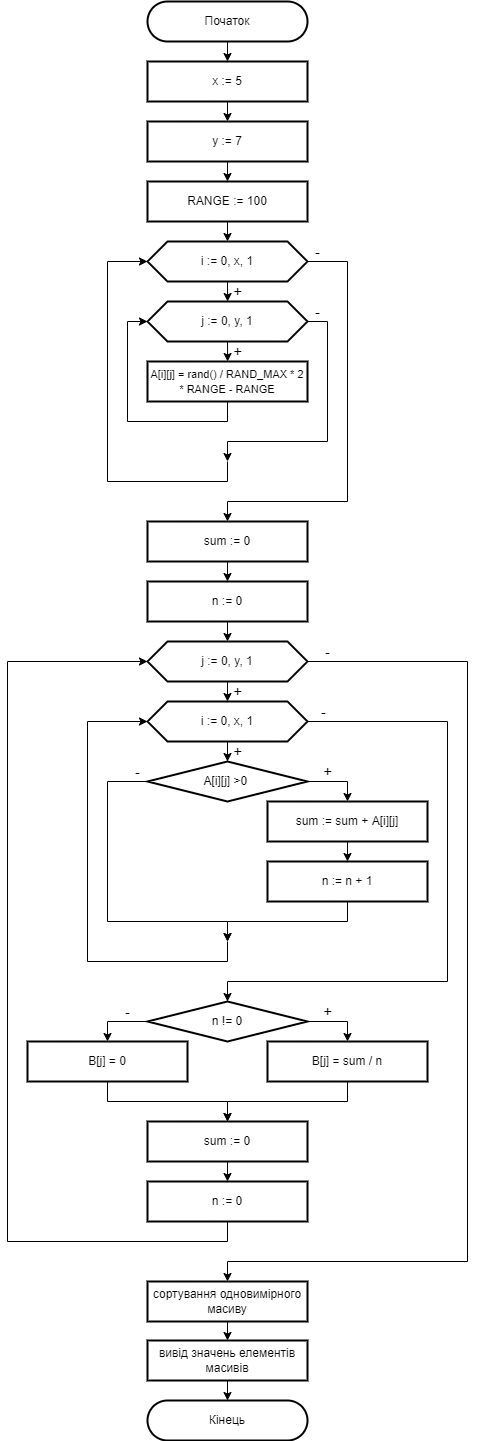
*Крок 2*

**

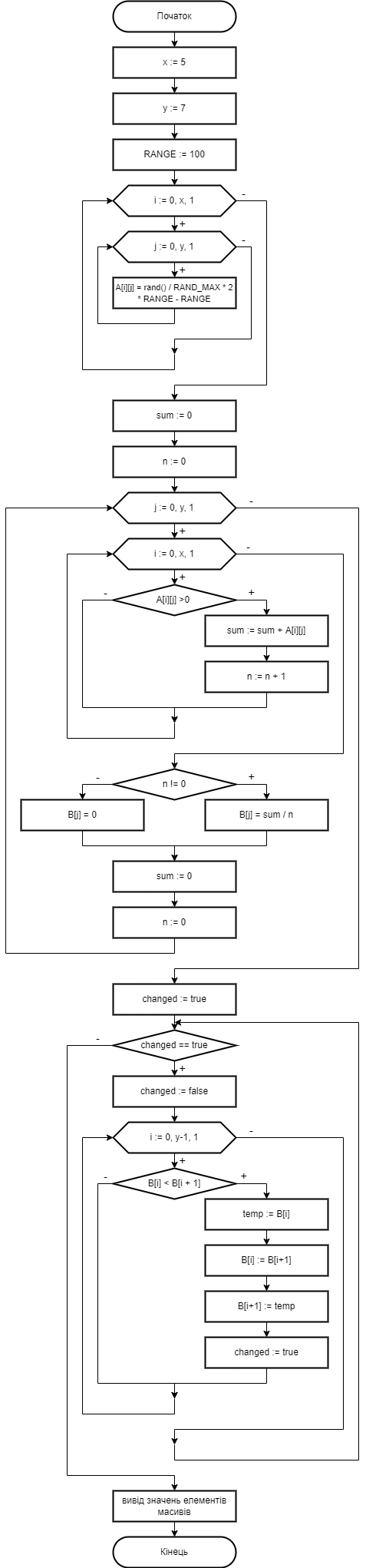
*Крок 3*

**

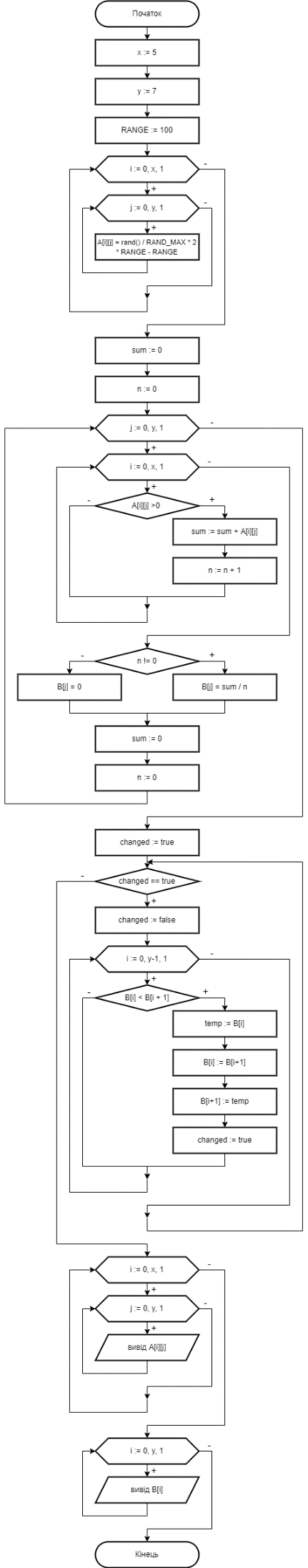
*Крок 4*

**

*Крок 5*

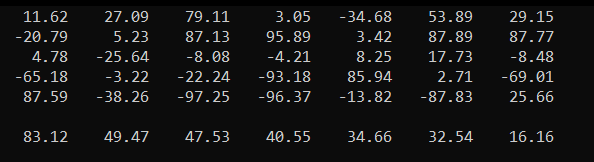
**

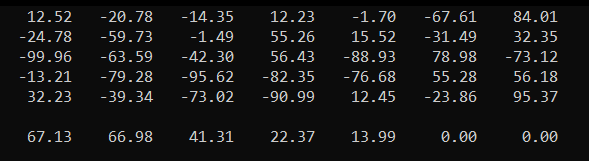
*Крок 6*

**

1. **Складання програми**
2. **Тестування програми**

Результат не залежить від вхідних даних, достатньо відеокопії результату





Програма відповідає поставленій задачі, помилок не виявлено.

1. **Висновки**

Під час виконання даної лабораторної роботи було розроблено й випробувано алгоритм ініціалізації двовимірної матриці обходом за рядками, її обробки обходом за стовпцями, сортування одновимірного масиву методом обміну, досліджено алгоритми пошуку та сортування, набуто практичних навичок використання цих алгоритмів під час складання програмних специфікацій.