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 № 6 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослiдження алгоритмiв обходу масивiв

»

Варіант 17

Виконав студент ІП-14 Макаревич Владислав Анатолійович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 202 1

**Лабораторна робота 9**

**Варіант 17**

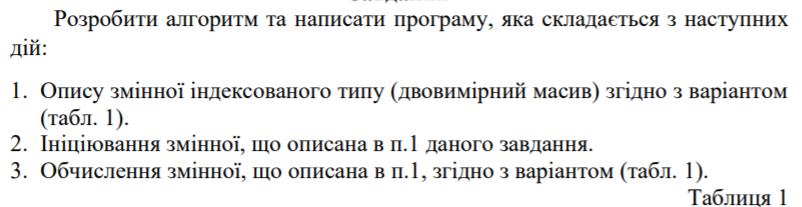
**Тема роботи:** Дослiдження алгоритмiв обходу масивiв

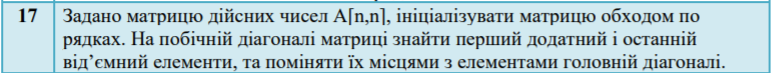
**Мета роботи:**

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

**Хід роботи**

**Завдання**

****

****

**Постановка задачі**

1 Опис змінної індексованого типу(двовимірний масив)

2 Ініціювання змінної, що описана

3 Обчислення змінної

Математична модель

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Ім’я | Тип | Призначення |
| 1 | array1 | Дійсний | Вхідне дане(Двовимірний масив) |
| 2 | arr | Дійсний | Вихідне дане(Двовимірний масив) |

Крок 1: Визначити основні дії

Крок 2: Заповнення масиву

Крок 3: Обробка масиву

Крок 4: Виведення

Псевдокод

Крок 1

Початок

Введення

Ініціалізація матриці, обробка масиву

Виведення

Кінець

Крок 2

Початок

size1=5

Функція FillArray( arr[][size1])

Для i = 0; i < size1; i++

Повторити

Для j = 0; j < size1; j++

Повторити

arr[i][j] = (rand() % 101 - 50) / 10.0

Все повторити

Все повторити

Виведення

Кінець

Крок 3

Початок

size1=5

Функція FillArray( arr[][size1])

Для i = 0; i < size1; i++

Повторити

Для j = 0; j < size1; j++

Повторити

arr[i][j] = (rand() % 101 - 50) / 10.0

Все повторити

Все повторити

Функція ChangeArray( arr[][size1])

bool positiveExict = false

bool negativeExict = false

Для i = 0, j = size1 - 1; i < size1 && j >= 0; i++, j—

повторити

Якщо arr[i][j] > 0

Якщо !positiveExict

То temp = arr[i][j];

arr[i][j] = arr[i][size1 - j - 1];

arr[i][size1 - j - 1] = temp;

positiveExict = true;

все якщо

інакше

якщо arr[i][j] < 0

то negativePositionI = i

negativePositionJ = j

все якщо

все якщо

якщо negativeExict

то temp = arr[negativePositionI][negativePositionJ];

arr[negativePositionI][negativePositionJ] = arr[negativePositionI][size1 - negativePositionJ - 1];

arr[negativePositionI][size1 - negativePositionJ - 1] = temp;

все якщо

все повторити

Виведення

Кінець

Крок 4

Початок

size1=5

Функція FillArray( arr[][size1])

Для i = 0; i < size1; i++

Повторити

Для j = 0; j < size1; j++

Повторити

arr[i][j] = (rand() % 101 - 50) / 10.0

Все повторити

Все повторити

Функція ChangeArray( arr[][size1])

bool positiveExict = false

bool negativeExict = false

Для i = 0, j = size1 - 1; i < size1 && j >= 0; i++, j—

повторити

Якщо arr[i][j] > 0

Якщо !positiveExict

То temp = arr[i][j];

arr[i][j] = arr[i][size1 - j - 1];

arr[i][size1 - j - 1] = temp;

positiveExict = true;

все якщо

інакше

якщо arr[i][j] < 0

то negativePositionI = i

negativePositionJ = j

все якщо

все якщо

якщо negativeExict

то temp = arr[negativePositionI][negativePositionJ];

arr[negativePositionI][negativePositionJ] = arr[negativePositionI][size1 - negativePositionJ - 1];

arr[negativePositionI][size1 - negativePositionJ - 1] = temp;

все якщо

все повторити

Функція ShowArray( arr[][size1])

Для i = 0; i < size1; i++

Повторити

Для j = 0; j < size1; j++

Повторити

Виведення arr[i][j]

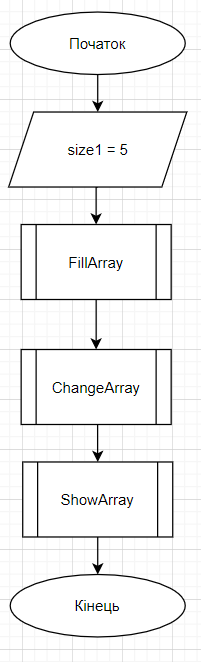
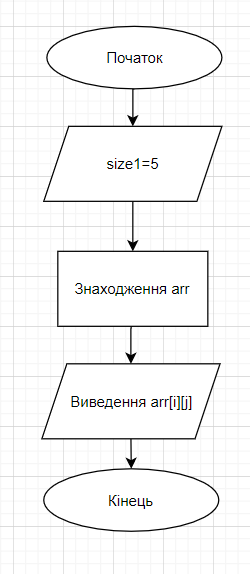
Все повторити

Все повторити

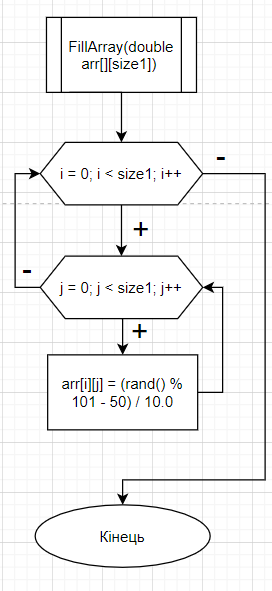
Кінець

Блок-схема

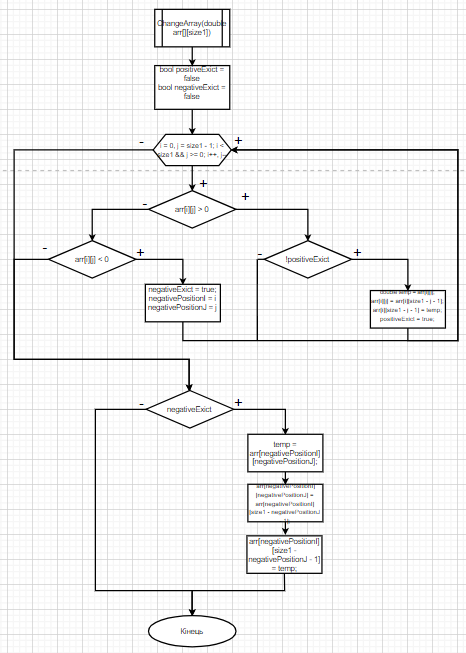
Крок 1



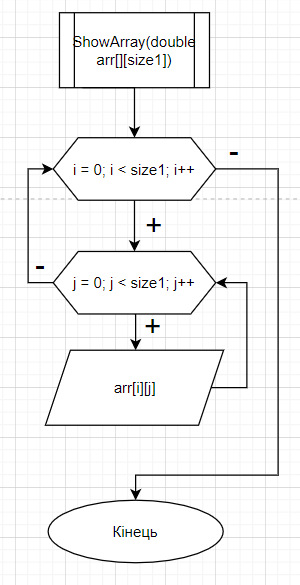
Крок 2



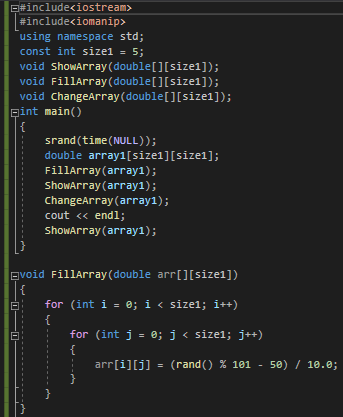
Крок 3

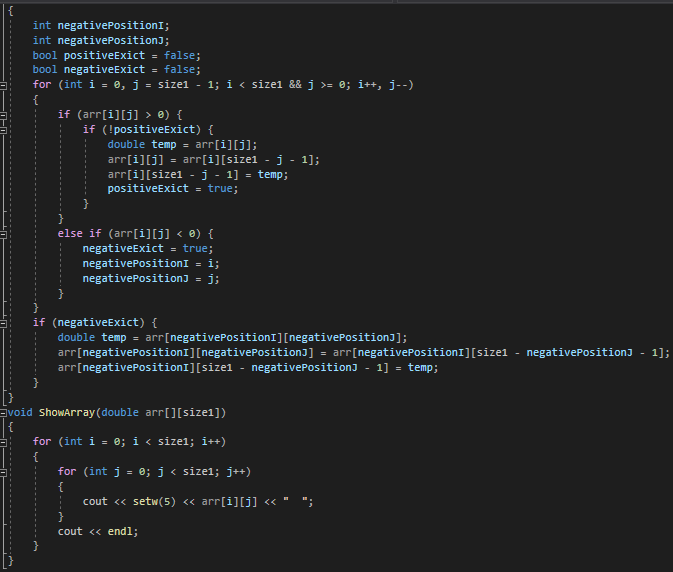


Крок 4

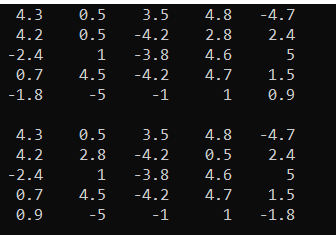


Код програми





Результат



Висновок: Досліджено алгоритм обходу масивів за допомогою циклів та функцій. Набуто практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В результаті виконання лабораторної роботи отримано алгоритм для обходу масиву по рядках, елементи головної діагоналі змінилися на перший додатній та останній від’ємний елемент побічної діагоналі. Для отримання результату потрібно було використати математичну модель, псевдокод, а також блок-схему. Робота написана на с++.