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 16

Виконав студент ІП-12 Ковинєв Владислав Андрійович

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.

Задача. Дана змінна індексованого типу (двовимірний масив) розміром 5х7. Треба ініціювати цю змину, у якій елементи є цілими числами, та створити нову змінну (одновимірний масив) та ініціювати її елементами, значенням яких буде дорівнювати значенню середнього арифметичного від`ємних значень елементів стовпців двовимірного масиву. Відсортувати їх за спаданням.

**I.** Постановка задачі. Створити двовимірний масив з довільними елементами. Для кожного з семи стовпців вирахувати середнє арифметичне від’ємних значень, створити з них новий одновимірний масив. У другому масиві відсортувати елементи обміном за спаданням.

**II.** Побудова математичної моделі

Складемо таблицю імен змінних



Складемо таблицю функцій



Таким чином математичне формулювання задачі зводиться до створювання матриці mx необхідного розміру (згідно вводу розміром rows стовпців на columns стовпців) за допомогою функції FillMatrix(cols, rows). Потім обчислення середнього арифметичного від’ємних елементів кожного стовпця матриці за допомогою функції CalcArithm(mx, columns, rows). Та внесення цього значення у масив розміром columns. Потім ми сортуємо отриманий масив методом бульбашки, а також виводимо проміжні дані та результат.

***Псевдокод***

**початок**

**Введення rows = 5, columns = 7, downRange = -99, upRange = 100**

matrixA: = FillMatrix (columns, rows)

OutputMatrix (matrixA, columns, rows)

arr: = CalcArithm (matrixA, columns, rows)

OutputArray (arr, columns)

BubbleSort (arr, columns)

OutputArray (arr, columns)

**кінець**

**підпрограма FillMatrix (int** **columns, int rows)**

a[rows]

*повторити для i від 0 до rows*

a[i][columns]

*все повторити*

*повторити для i від 0 до rows*

*повторити для j від 0 до columns*

a[i][j] := Random(downRange, upRange)

*все повторити*

*все повторити*

**повернути** a

**все підпрограма**

**підпрограма OutputMatrix (int\*\* matrix, int columns, int rows)**

*повторити для i від 0 до rows*

*повторити для j від 0 до columns*

**Виведення matrix[i][j]**

*все повторити*

*все повторити*

**все підпрограма**

**підпрограма CalcArithm (int\*\* matrix, int columns, int rows)**

arr[columns]

sum := 0

num := 0

*повторити для j від 0 до columns*

*повторити для i від 0 до rows*

*якщо matrix[i][j] < 0*

sum += matrix[i][j]

num++

*все якщо*

*все повторити*

*якщо num != 0*

arr[j] := sum/num

*інакше*

arr[j] := 0

*все якщо*

sum := 0

num := 0

*все повторити*

**повернути arr**

**все підпрограма**

**підпрограма OutputArray (int\* arr, int columns, int rows)**

*повторити для i від 0 до rows*

**Виведення arr[i]**

*все повторити*

**все підпрограма**

**підпрограма BubbleSort (int\* arr, int size)**

tmp := 0

*повторити для i від 0 до size*

*повторити для j від 0 до size*

*якщо arr[i] < arr[j]*

tmp = arr[i]

arr[i] = arr[j]

arr[j] = tmp

*все якщо*

*все повторити*

*все повторити*

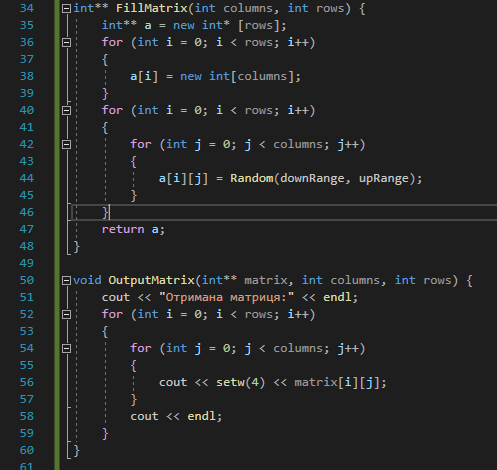
**все підпрограма**

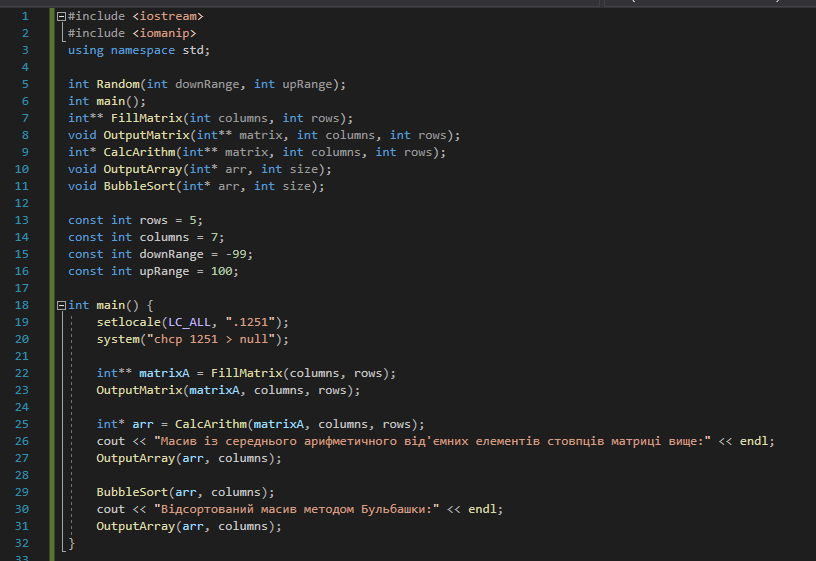
**підпрограма Random (int downRange, int upRange)**

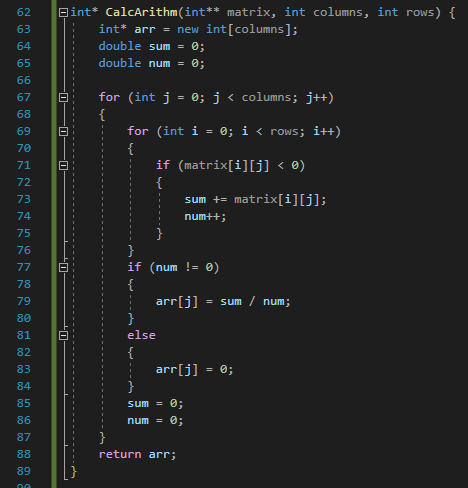
**return rand() % (upRange - downRange + 1) + downRange**

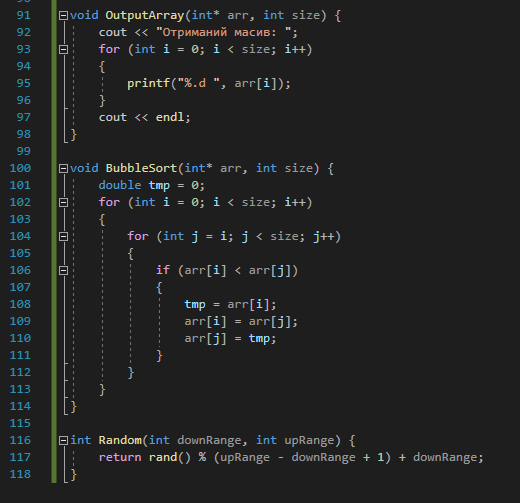
**все підпрограма**

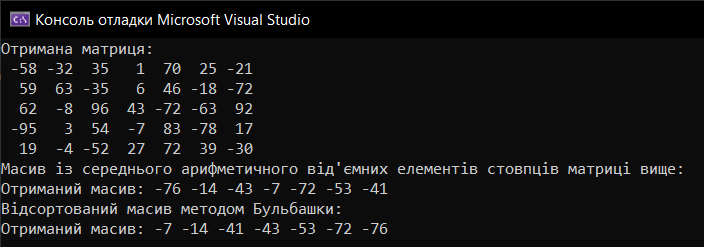
      

**IV.** Випробування алгоритму









**V.** Висновок

Було досліджено алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В результаті виконання лабораторної роботи було ініційовано двовимірний масив, для кожного стовпця стовпця розраховано середнє арифметичне, отримані значення збережено у другому масиві та відсортовано методом обміну, розділивши задачу на 3 кроки: визначення основних дій, деталізація генерації другого масиву за допомогою підпрограми, сортування елементів другого масиву