

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 14

Виконав студент: ПП-15 Кондрацька Соня Леонідівна

Перевірів: Вечерковська Анастасія Сергіївна

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 14

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

14	4 x 6	Дійсний	Із добутку від'ємних значень елементів стовпців двовимірного масиву. Відсортувати методом бульбашки за спаданням.
-----------	--------------	----------------	--

1) Постановка задачі

Ініціюємо двовимірний масив дійсного типу розміром 4x6. Заносимо в масив змінні. Створюємо одновимірний масив дійсного типу. Заносимо в нього добутки від'ємних елементів кожного стовпця. За допомогою формул метода бульбашки сортуємо елементи масиву за спаданням.

2) Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Дійсний	arr	Вхідні дані
Одновимірний масив	Дійсний	b	Вихідні дані
Двовимірний масив у підпрограмах	Дійсний	a	Вхідні дані
Одновимірний масив у підпрограмах	Дійсний	b_f	Вихідні дані
Змінна для сортування	Дійсний	tmp	Проміжні дані
Змінна для визначення добутку	Дійсний	kol	Проміжні дані

в функції *input()* в масив *arr* заносимо елементи за допомогою функції *rand()*. За допомогою підпрограми *func()* знаходимо добуток від'ємних елементів кожного стовпця *i* заносимо в одновимірний масив *b*. Сортуємо цей масив за спаданням методом бульбашки у функції *sort()*.

3) Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Викликаємо першу підпрограму

Крок 3. Викликаємо другу підпрограму

Крок 4. Викликаємо третю підпрограму

Псевдокод

Основна програма:

Крок 1

Початок

Ініціювання двох масивів

Виклик першої підпрограми

Виклик другої підпрограми

Виклик третьої підпрограми

Кінець

Крок 2

Початок

double arr[4][6], b[6];

input(arr);

func(arr, b);

sort(b);

Кінець

Підпрограми:

input(double a)

для і від 0 до 4 повторити

для j від 0 до 6 повторити

a[i][j]=double(rand())/RAND_MAX*200+-100;

все повторити

все повторити

Кінець

func(double a[4][6], double b_f[6])

для j від 0 до 6 повторити

kol=1;

для і від 0 до 4 повторити

якщо a[i][j]<0 то

kol*=a[i][j];

все якщо

b_f[j]=kol;

все повторити

все повторити

Кінець

sort(double b_f[6])

double tmp;

для і від 0 до 4 повторити

для j від 0 до 6 повторити

якщо b_f[j+1]>b_f[j] то

tmp = b_f[j + 1];

b_f[j + 1] = b_f[j];

b_f[j] = tmp;

все якщо

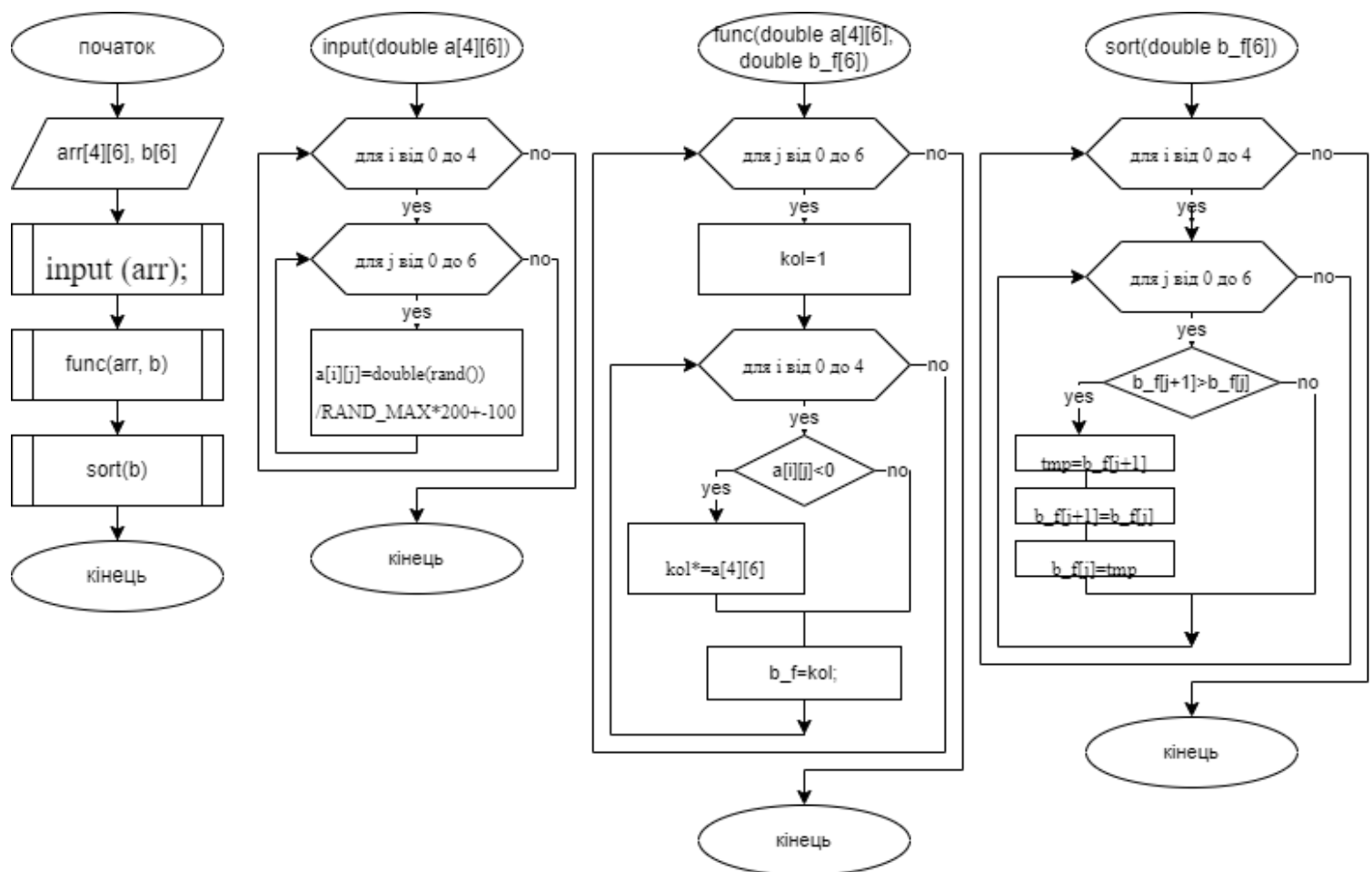
все повторити

все повторити

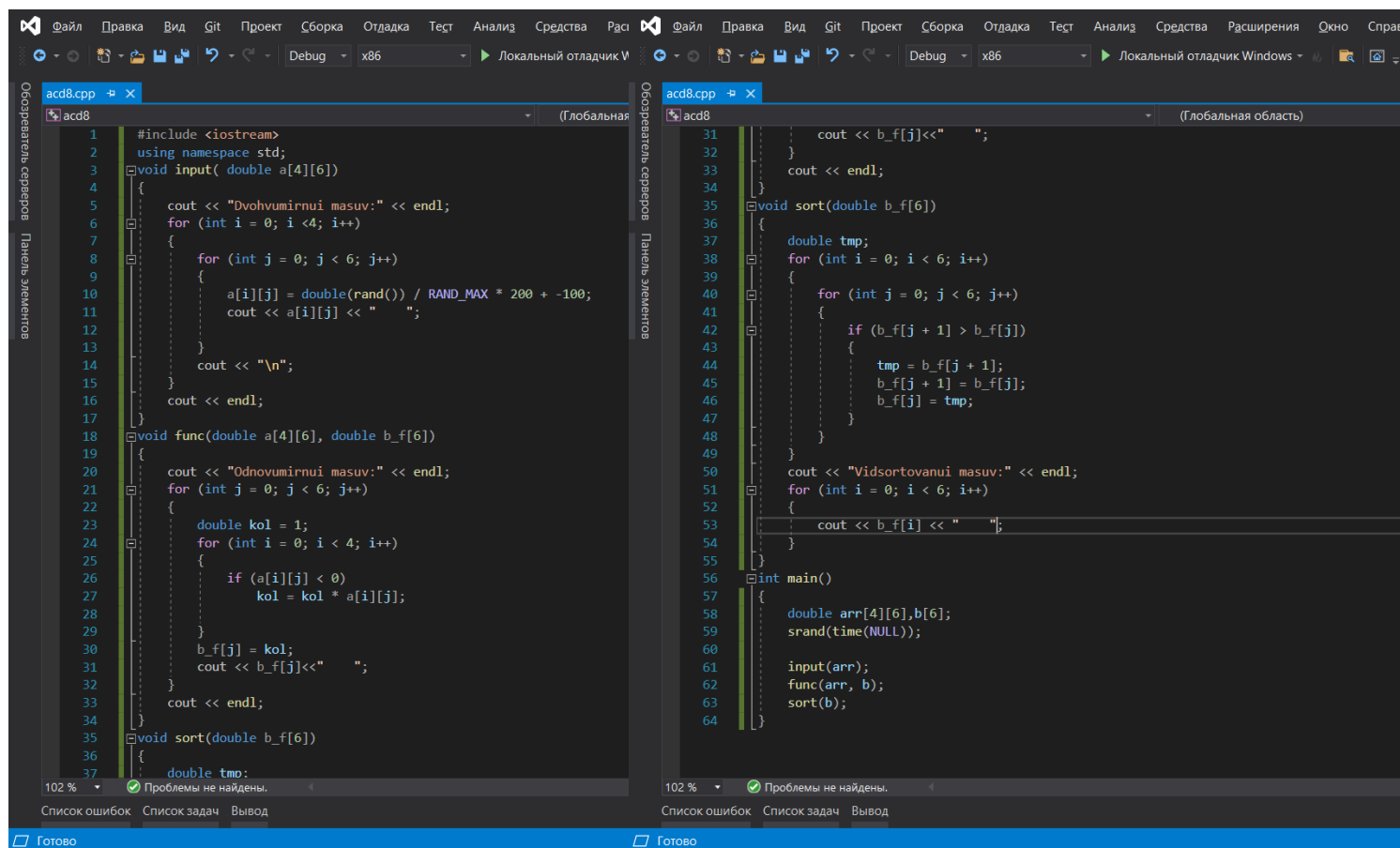
Кінець

4) Блок-схема

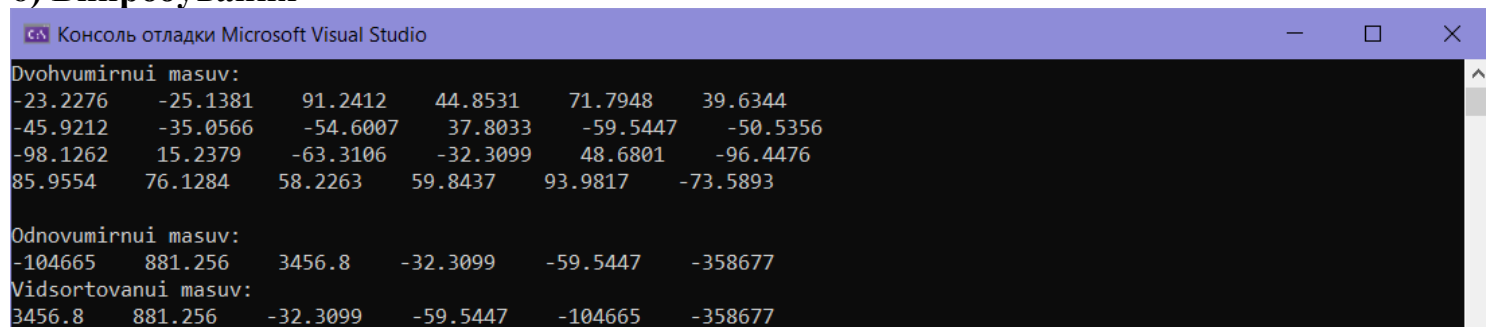
Основна програма: Підпрограми:



5) Код



6) Випробування



7) Висновки

Ми дослідили алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

В результаті виконання лабораторної роботи ми отримали програму для заповнення двовимірного масиву, заповнення одновимірного масиву, сортування масиву методом бульбашки, розділивши задачу на 4 кроків: визначення основних дій, виклик першої

підпрограми, виклик другої підпрограми, виклик третьої підпрограми.