

*Додаток 1*

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки  
Кафедра автоматизованих систем обробки інформації  
і управління

Звіт

з лабораторної роботи № 8  
з дисципліни «Алгоритми та структури даних-1. Основи  
алгоритмізації»  
«Дослідження алгоритмів пошуку та сортування »  
Варіант 8

Виконав

\_\_\_\_\_  
ІІ-15, Дацьо Іван Іванович

студент

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна

( 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.

#### Індивідуальне завдання :

#### Варіант 8

##### Завдання

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).

8	4 x 6	Цілий	Із максимальних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.
---	-------	-------	--

### 1. Постановка задачі.

Створити двовимірний масив  $4 \times 6$  цілих значень змінних. Створити одновимірний масив із максимальних значень елементів стовпців двовимірних масивів. Відсортувати цей масив обміном за спаданням. Результатом є виведення відсортованого масиву із максимальних значень стовпців двовимірного масиву .

## 2. Побудова математичної моделі

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Кількість рядків матриці	Цілий	n	Початкові дані
Кількість стовпців матриці	Цілий	m	Початкові дані
Лічильник1 циклу	Цілий	i	Параметр циклу
Лічильник2 циклу	Цілий	j	Параметр циклу
Двовимірний масив	Цілий	array	Проміжні дані
Одновимірний масив	Цілий	b	Проміжні/кінцеві дані
Максимальний елемент стовпця	Цілий	max	Проміжні дані
Змінна сортування	Цілий	tmp	Проміжні дані
Присвоєння значень двовимірному масиву	Процедура	input	Процедура
Вивід двовимірного масиву	Процедура	output	Процедура
Новий масив	Процедура	n_arr	Процедура
Сортування одновимірного масиву	Процедура	sort	Процедура

Індексацію масиву розпочинаємо із 0 .

Генерацію випадкових чисел позначаємо rand , для генерації від від -100 до 400 :  
 rand() % 500 -100

Математичне формулювання задачі зводиться до таких дій :

Випадковим чином генеруємо значення масиву array в функції **input** з допомогою двох циклів . Генерування випадкових значення відбуватиметься за допомогою класу Rand . Наступною дією буде виведення цього двовимірного масиву , для виразності в функції **output** за допомогою двох циклів .Наступною дією буде створення нового масиву у функції **n\_arr** . Для цього створюємо два цикла , вкладем

з яких є цикл рядків , у зовнішньому циклі створюємо змінну *max* , якій присвоюємо значення першого елемента стовпця і змінюватимемо його якщо він буде більшим за наступне значення в цьому стовпці . Присвоюємо новому масиву значення максимальних елементів , у відповідності до номеру стовпця. Виводимо його для виразності . В функції **sort** відсортовуємо утворений масив одним із видів сортування обміном – бульбашкою , за спаданням. Виводимо масив для виразності.

### 3.Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми .

Крок 1. Визначимо основні дії .

Крок 2.Деталізуємо дію генерацію двовимірного масиву за допомогою функції.

Крок 3.Деталізуємо дію виведення двовимірного масиву за допомогою функції .

Крок 4. Деталізуємо дію створення нового масиву.

Крок 5. Деталізуємо дію сортування нового масиву за допомогою функції.

### 4. Псевдокод

#### Основна програма :

початок

`n = 4`

`m = 6`

`array [4][6], b[6]`

`input ( n , m , array )`

`output ( n , m , array )`

`n_array ( n , m , array , b)`

`sort(b)`

кінець

#### Підпрограми :

`input ( n , m , array )`

для *i* від 0 до *n* повторити

**для j від 0 до m повторити**

**array [i][j] = rand () % 500 - 100**

**все повторити**

**все повторити**

**кінець**

**output ( n , m , array )**

**для i від 0 до n повторити**

**для j від 0 до m повторити**

**Вивести array[i][j]**

**все повторити**

**все повторити**

**кінець**

**n\_array ( n , m , array , b)**

**для j від 0 до m повторити**

**max = array [0][j]**

**для i від 0 до n повторити**

**якщо array [i][j] > max**

**то max = array [i][j]**

**все повторити**

**b[j] = max**

**Вивести b[j]**

**все повторити**

**кінець**

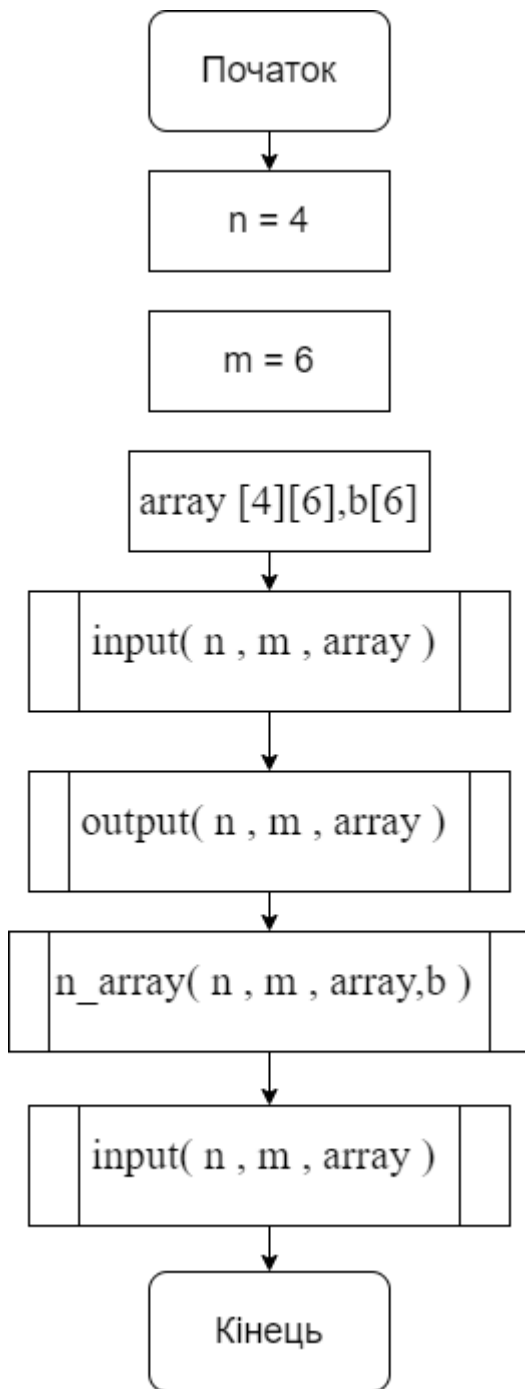
**sort(b)**

**для i від 0 до b повторити**

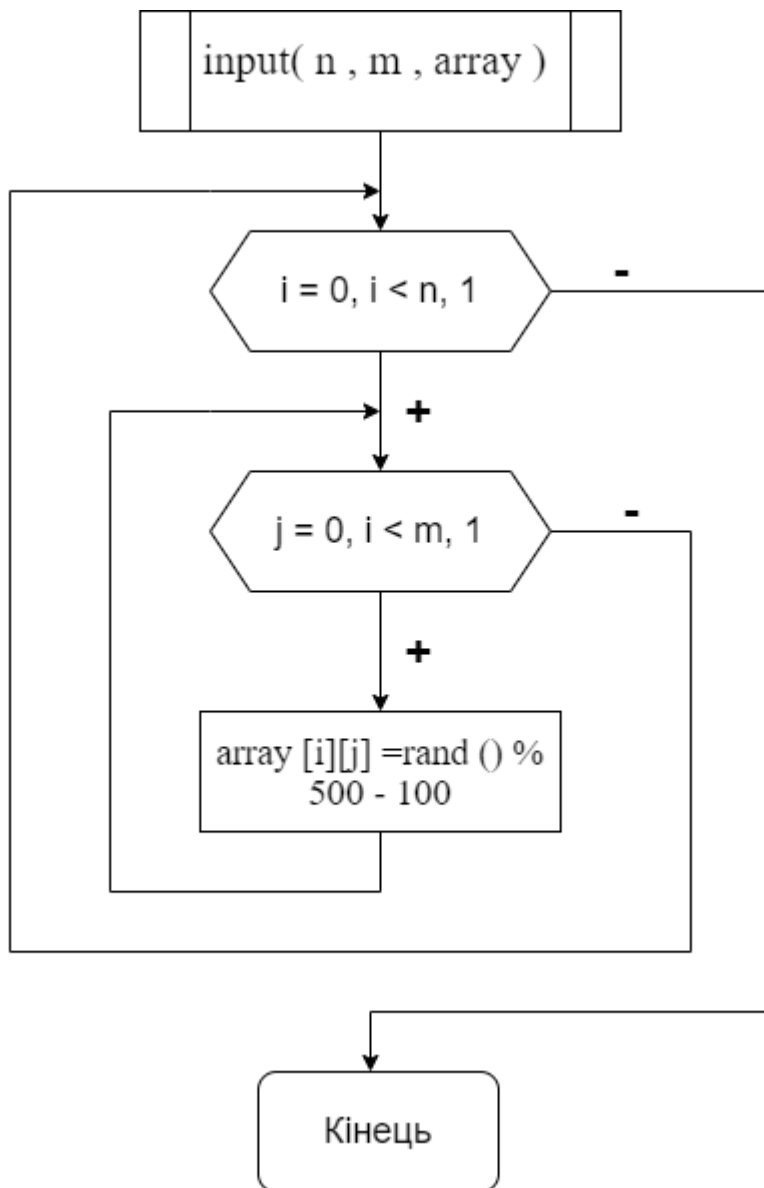
```
    для j від 0 до 6 повторити
        якщо  $b[j+1] > b[j]$ 
            то  $tmp = b[j + 1]$ 
                 $b[j + 1] = b[j]$ 
                 $b[j] = tmp$ 
        все якщо
    все повторити
для i від 0 до 6 повторити
    Вивести  $b[i]$ 
все повторити
кінець
```

## 5. Блок-схема алгоритму

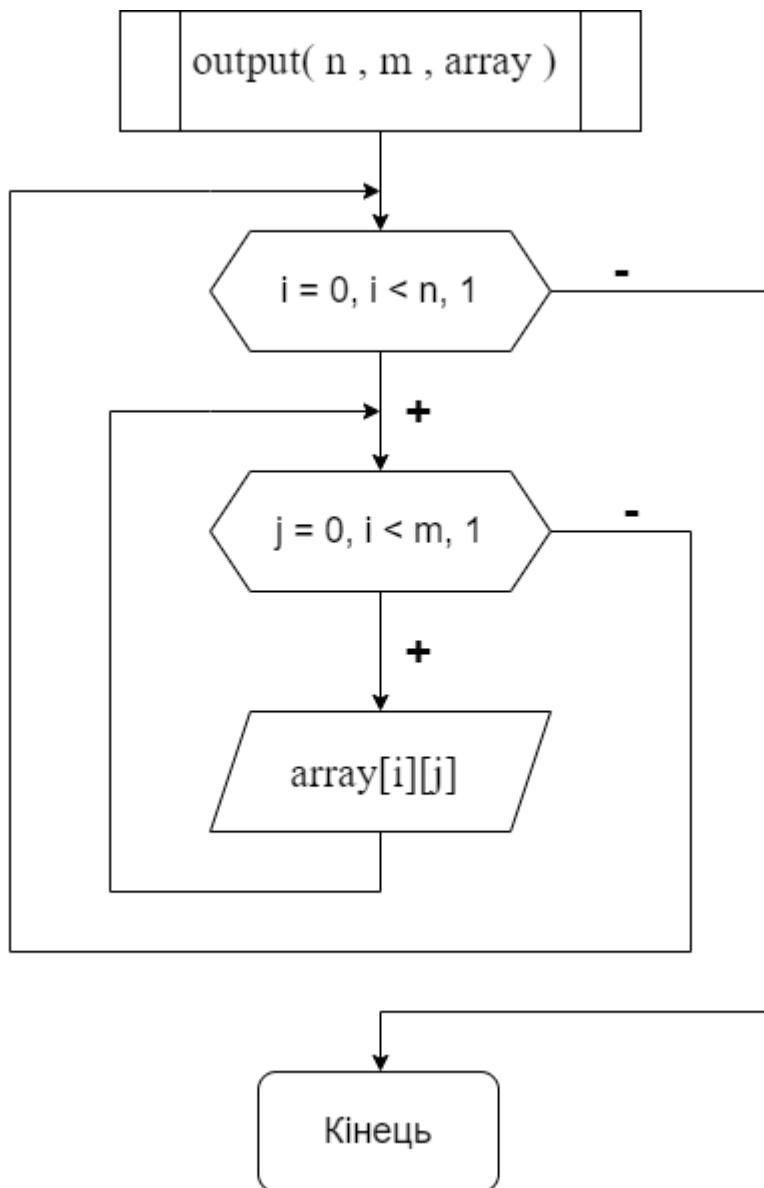
Основна програма :

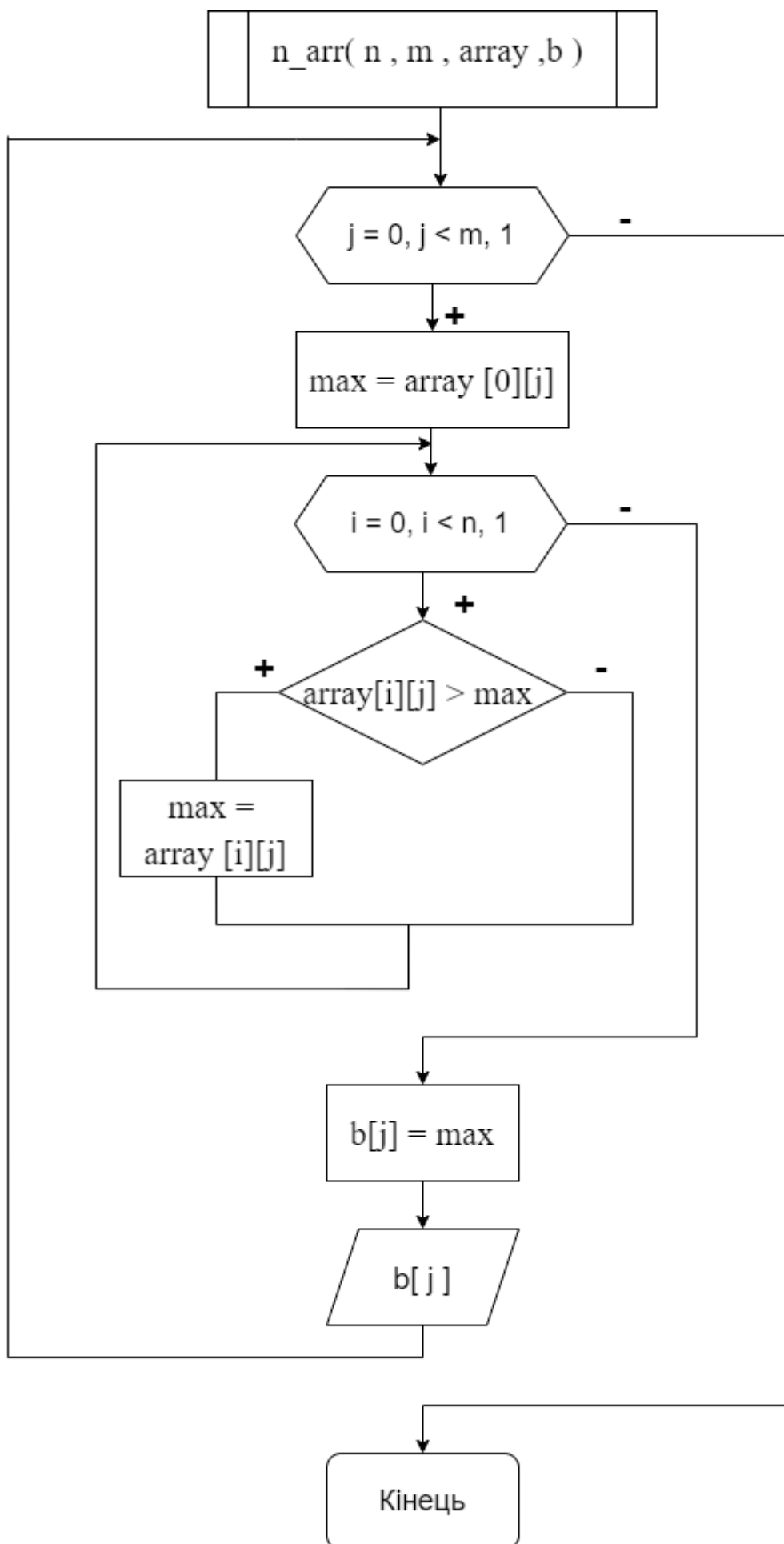


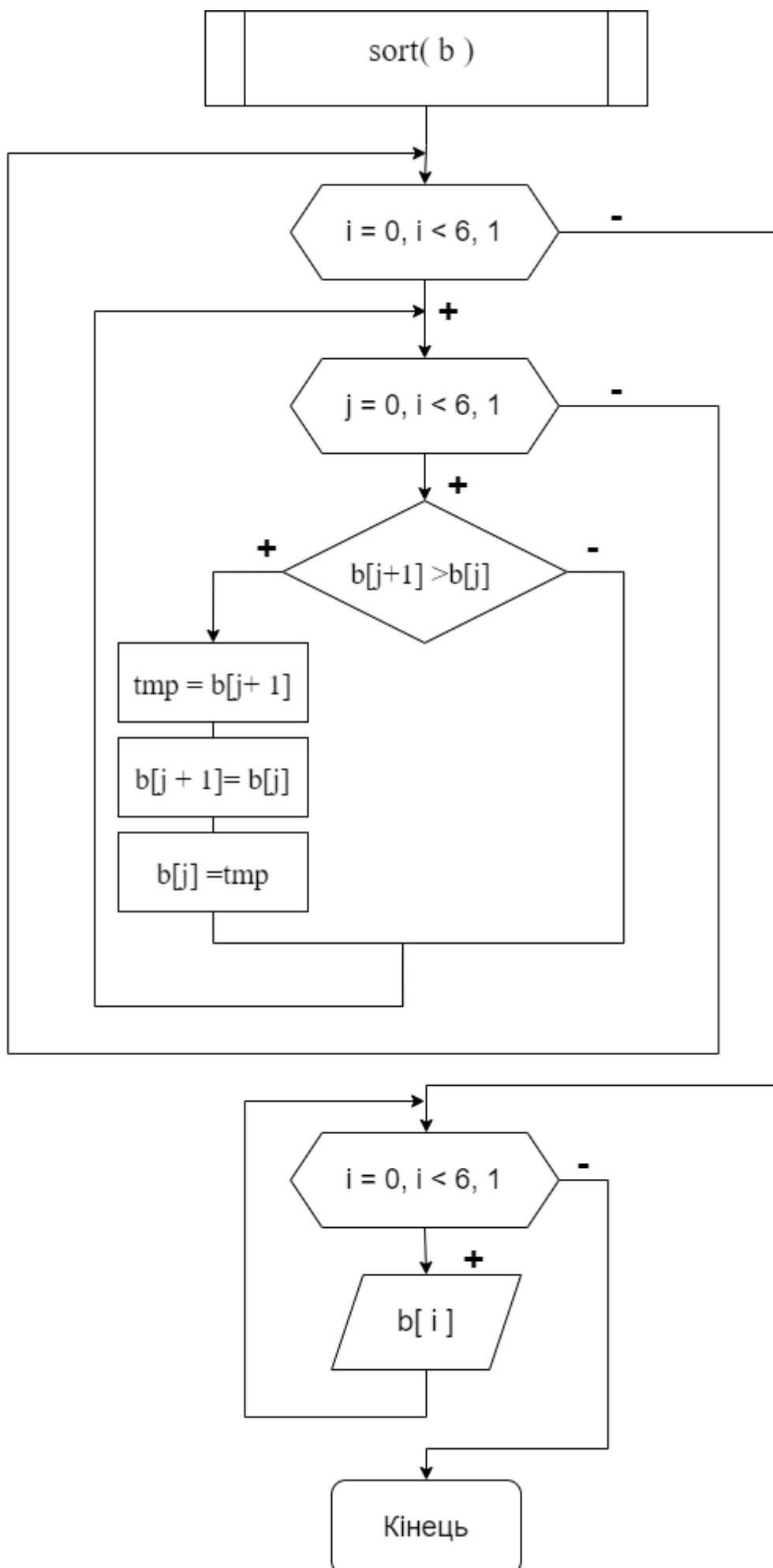
Підпрограми :











## 6. Код програми(c++)

```
1  #include <iostream>
2  #include <stdlib.h>
3  using namespace std ;
4  void input ( int n , int m , int array[4][6] ){
5      for (int i = 0; i < n ; i++){
6          for (int j = 0; j < m ; j++){
7              array[i][j] = rand()%500 + -100;
8          }
9      }
10 }
11 void output (int n , int m , int array[4][6] ){
12     for (int i = 0; i < n ; i++){
13         for (int j = 0; j < m ; j++){
14             cout << array[i][j] <<"\t";
15         }
16         cout << endl;
17     }
18 }
19 void n_arr ( int n , int m , int array [4][6] , int b[6]){
20     cout <<"Масив із максимальних значень стовпця:"<< endl;
21     for ( int j = 0; j < m; j++){
22         int max = array [0][j];
23         for ( int i = 1; i < n; i++){
24             if ( array[i][j] > max){
25                 max = array[i][j];
26             }
27         }
28         b[j] = max;
29         cout << b[j] <<"\t";
30     }
31 }
```

```

32 void sort (int b[6]){
33     cout << endl;
34     cout << "Відсортований масив:" << endl;
35     int tmp ;
36     for ( int i = 0; i < 6; i++){
37         for ( int j = 0; j < 6; j++){
38             if (b[j + 1] > b[j]){
39                 tmp = b[j + 1];
40                 b[j + 1] = b[j];
41                 b[j] = tmp;
42             }
43         }
44     }
45     for (int i = 0; i < 6; i++){
46         cout << b[i] << "\t";
47     }
48 }
49 int main(){
50     setlocale(LC_ALL, "Ukrainian");
51     srand(time(NULL));
52     int n = 4;
53     int m = 6;
54     int array[4][6], b[6];
55     input ( n , m , array );
56     output ( n , m , array );
57     n_arr ( n , m , array , b);
58     sort (b);
59 }

```

### Випробування :

```

140      -23      109      215      -65      -92
127      267      240      168      191      255
361      326      203      204      97       353
346      91       -95      -58      -52      398
Масив із максимальних значень стовпця:
361      326      240      215      191      398
Відсортований масив:
398      361      326      240      215      191

```

## 7.Висновок

Ми дослідили алгоритми пошуку та сортування, набути практичних навичок використання під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали програму для заповнення двовимірного масиву, пошуку найбільшого елемента стовпця, заповнення одновимірного масиву, сортування одновимірного масиву методом обміну , а саме методом бульбашки розділивши задачу на 4 кроки: визначення основних дій, генерації двовимірного масиву за допомогою функції, виведення двовимірного масиву за допомогою функції б створення нового масиву , сортування нового масиву за допомогою функції .