
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 18

Виконав студент: ІП-15 Лазьов Кирило Владиславович

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 18

Постановка задачі

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).

18	5 x 5	Дійсний	Із від'ємних значень елементів побічної діагоналі двовимірного масиву. Відсортувати методом вставки за спаданням.
----	-------	---------	---

Змінна	Тип	Ім'я	Призначення
Двовимірний масив	Дійсний	matrix	Вхідні дані
Одновимірний масив	Дійсний	array	Результат
Кількість елементів масиву	Натуральний	k	Проміжні дані
Змінна для заповнення масиву	Натуральний	a	Проміжні дані
Змінна для зберігання попереднього значення масиву	Дійсний	t	Проміжні дані
Параметр циклу	Натуральний	i	Проміжні дані
Параметр циклу	Натуральний	j	Проміжні дані

Розв'язання

Двовимірний масив `matrix` розмірністю 5x5 заповнюється випадковими числами за допомогою двох арифметичних циклів у функції `matrix_input`. У функції `array_input` перебираємо елементи побічної діагоналі двовимірного масиву, від'ємні значення заносимо у одновимірний масив `array` і повертаємо значення кількості елементів масиву (змінна `k`). Далі у функції `array_sort` переставляємо елементи масиву `array` у порядку спадання за допомогою двох арифметичних циклів. Виводимо масив за допомогою арифметичного циклу.

Крок 1. Визначаємо основні дії

Крок 2. Виклик першої підпрограми

Крок 3. Виклик другої підпрограми

Крок 4. Виклик третьої підпрограми

Крок 5. Виведення отриманого масиву

Псевдокод

Основна програма:

```
matrix[5][5]
```

```
array[5]
```

```
k = 0
```

```
matrix_input(matrix)
```

```
k = array_input(matrix, array, k)
```

```
array_sort(array, k)
```

```
    повторити для i від 0 до k
```

```
        виведення array[i]
```

```
    все повторити
```

```
Кінець
```

Підпрограми:

```
matrix_input(double matrix[5][5])
```

```
Початок
```

```
    повторити для i від 0 до 5
```

```
        повторити для j від 0 до 5
```

```
            matrix[i][j] = double(rand())/RAND_MAX*200-100
```

```
        все повторити
```

```
    все повторити
```

```
Кінець
```

array_input(double matrix[5][5], array[5])

Початок

j = 4

a = 0

повторити для i від 0 до 5

якщо matrix[i][j]<0

то

array[a]=matrix[i][j]

k++

a++

j --

інакше

j --

все повторити

повернути k

Кінець

array_sort(double array[5], int k)

повторити для i від 0 до k

повторити для j від 0 до i

якщо array[i]>array[i-j]

то

t=array[i-j]

array[i-j]=array[i]

array[i]=t

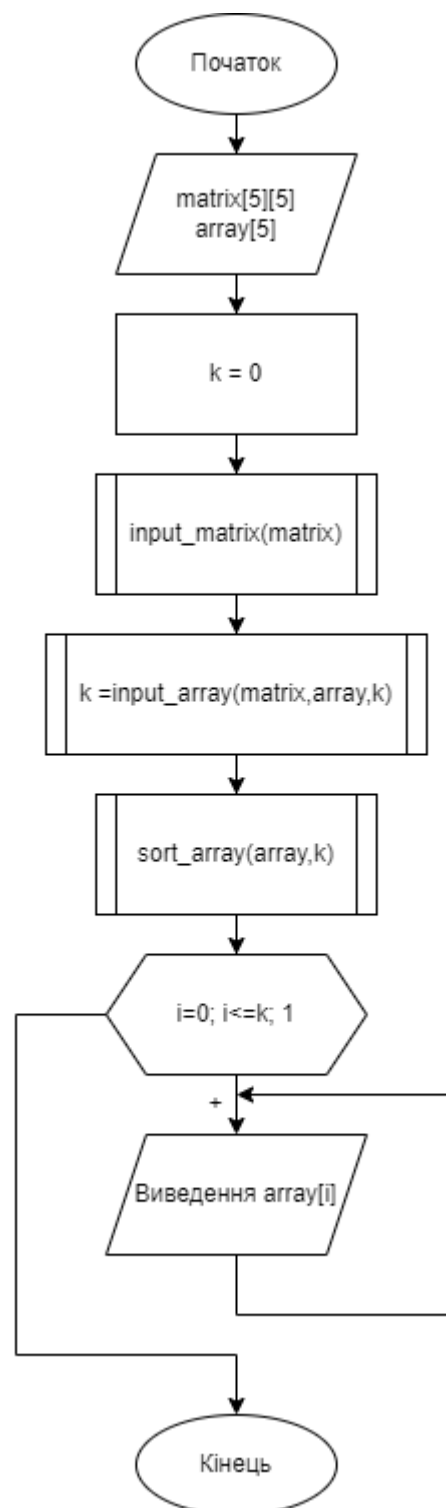
все повторити

все повторити

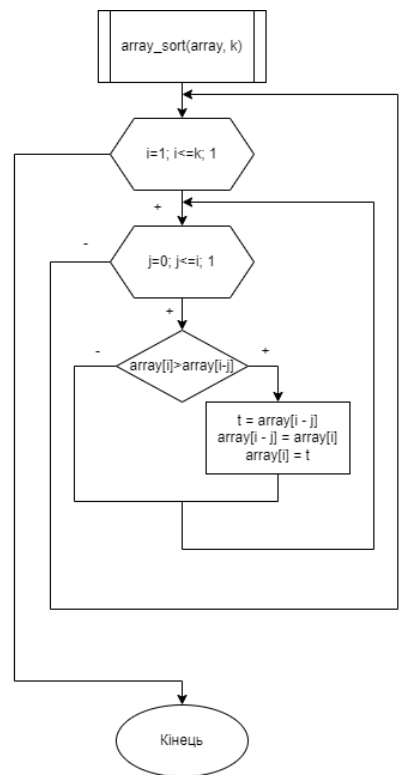
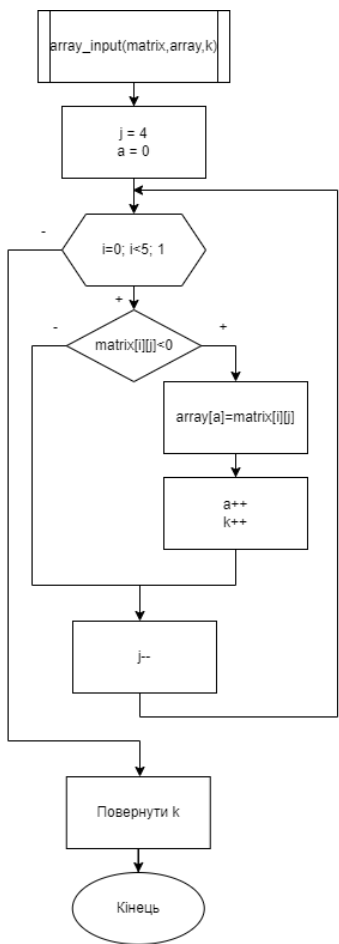
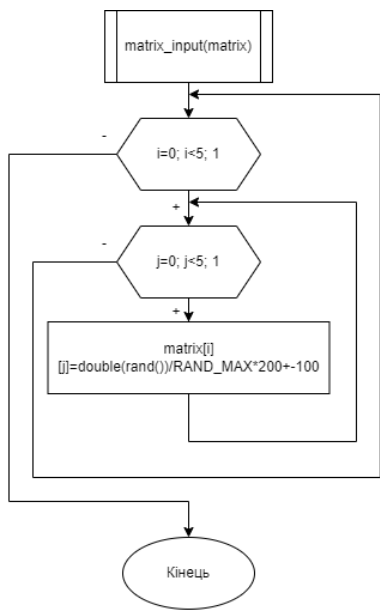
Кінець

Блок-схема

Основна програма:



Підпрограми:



Код програми

```
1  #include <iostream>
2  #include<time.h>
3  using namespace std;
4  void matrix_input(double matrix[5][5]) {
5      cout << "Матриця:" << endl;
6      for (int i = 0; i < 5; i++) {
7          for (int j = 0; j < 5; j++) {
8              matrix[i][j] = double(rand()) / RAND_MAX * 200 + -100;
9              cout << matrix[i][j] << " ";
10         }
11         cout << endl;
12     }
13 }
14 float array_input(double matrix[5][5], double array[5], int k) {
15     cout << "Масив:" << endl;
16     int j = 4;
17     int a = 0;
18     for(int i = 0; i < 5; i++) {
19         if (matrix[i][j] < 0) {
20             array[a] = matrix[i][j];
21             cout << array[a] << " ";
22             k++;
23             a++;
24             j--;
25         }
26         else {
27             j--;
28         }
29     }
30     cout << endl;
31     return k;
32 }
33 void array_sort(double array[5], int k) {
34     cout << "Відсортований масив:" << endl;
35     for (int i = 1; i <= k; i++) {
36         for (int j = 0; j <= i; j++) {
37             if (array[i] > array[i - j]) {
38                 float t = array[i - j];
39                 array[i - j] = array[i];
40                 array[i] = t;
41             }
42         }
43     }
44 }
45 }
46 }
47 int main()
48 {
49     setlocale(LC_ALL, "Ukrainian");
50     int k = 0;
51     double matrix[5] [5];
52     double array[5];
53     srand(time(NULL));
54
55     matrix_input(matrix);
56     k = array_input(matrix, array, k);
57     array_sort(array, k);
58
59     for (int i = 0; i != k; i++) {
60         cout << array[i] << " ";
61     }
62 }
63
64
```

Випробування

```
Матриця:  
73.4367 -69.4876 -20.1575 -65.1234 -33.5002  
64.1469 45.5061 22.9896 -7.41905 77.575  
41.5937 -8.34681 17.7526 2.68868 81.8842  
-28.8858 -63.4693 0.234993 51.4512 -95.6786  
25.6447 23.2948 41.142 -31.5775 -95.4466  
Масив:  
-33.5002 -7.41905 -63.4693  
Відсортований масив:  
-7.41905 -33.5002 -63.4693
```

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 від'ємними значеннями другорядної діагоналі двовимірного масиву, сортування масиву за спаданням.