
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 № 7 з дисципліни
«Алгоритми та структури даних-1. Основи алгоритмізації»
«Дослідження лінійного пошуку в послідовностях»

Варіант 18

Виконав студент: ІП-15 Лазьов Кирило Владиславович

Перевірів: Вечерковська Анастасія Сергіївна

Лабораторна робота 7

Дослідження складних рекурсивних алгоритмів

Мета – дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

Варіант 18

Постановка задачі

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису трьох змінних індексованого типу з 10 символьних значень.
2. Ініціювання двох змінних виразами згідно з варіантом.
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

18	$60 - 2 * i$	$40 + 3 * i$	Перше входження елемента з кодом 52
----	--------------	--------------	-------------------------------------

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	a	Вхідні дані
Другий масив	Символьний	b	Вхідні дані
Довжина масиву	Натуральний	length	Вхідні дані
Третій масив	Символьний	c	Проміжні дані
Індекс c	Натуральний	index	Проміжні дані
Індекс першого елемента масиву c з кодом 58	Натуральний	ind	Результат

Розв'язання

Генеруємо значення масивів a і b за допомогою функції get_ab(), використовуючи дані в умові формули. У підпрограмі get_c, за допомогою циклу з перевіркою на еквівалентність значень a та b, заповнюється масив c (входять тільки еквівалентні значення). Далі використовуючи функцію get_ind, знаходимо індекс першого елемента масиву c, з кодом 52. В результаті буде виведено отриманий індекс.

Крок 1. Визначаємо основні дії

Крок 2. Деталізуємо заповнення масивів за допомогою підпрограми get_ab

Крок 3. Деталізуємо знаходження елементів масиву с функцією get_c

Крок 4. Деталізуємо знаходження індексу елемента масиву с з кодом 52

Псевдокод

Крок 1

Початок

ind = 0

length = 10

Деталізуємо заповнення масивів за допомогою підпрограми get_ab

Деталізуємо знаходження елементів масиву функцією get_c

Деталізуємо знаходження індексу елемента масиву с з кодом 52

Виведення ind

Кінець

Крок 2

Початок

ind = 0

length = 10

get_ab(a, b, length)

Деталізуємо знаходження елементів масиву функцією get_c

Деталізуємо знаходження індексу елемента масиву с з кодом 52

Виведення ind

Кінець

Крок 3

Початок

ind = 0

length = 10

get_ab(a, b, length)

get_c(a, b, c, length)

Деталізуємо знаходження індексу елемента масиву с з кодом 52

Виведення ind

Кінець

Крок 4

Початок

length = 10

ind = 0

get_ab(a, b, length)

get_c(a, b, c, length)

ind = get_ind(c, ind, length)

Виведення ind

Кінець

Підпрограми:

get_ab(a, b, length)

Початок

повторити для i від 0 до length - 1

$a[i] = 60 - 2 * i$

$b[i] = 40 + 3 * i$

все повторити

Кінець

get_c(a, b, c, length)

Початок

index = 0

повторити для i від 0 до length - 1

повторити для j від 0 до length - 1

якщо $a[i] == b[j]$ то

$c[index] = a[i]$

index++

все якщо

все повторити

все повторити

Кінець

get_ind(c, ind, length)

Початок

повторити для i від 0 до length - 1

якщо $c[i] == 52$ то

ind = i

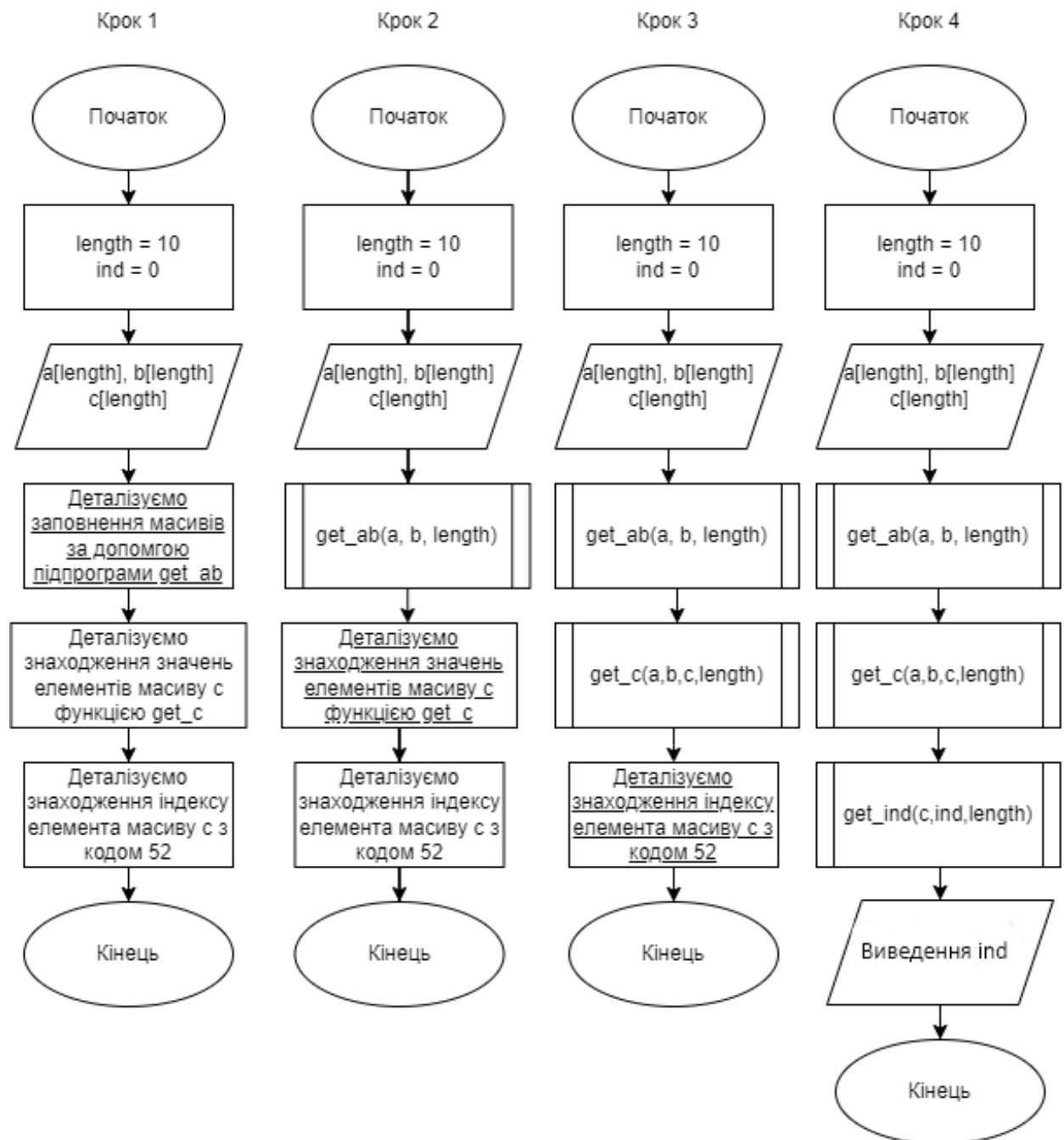
Повернути ind

все якщо

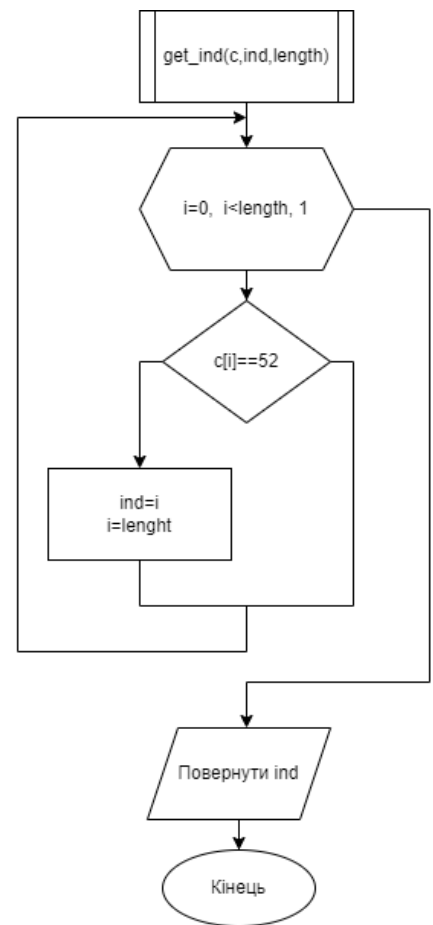
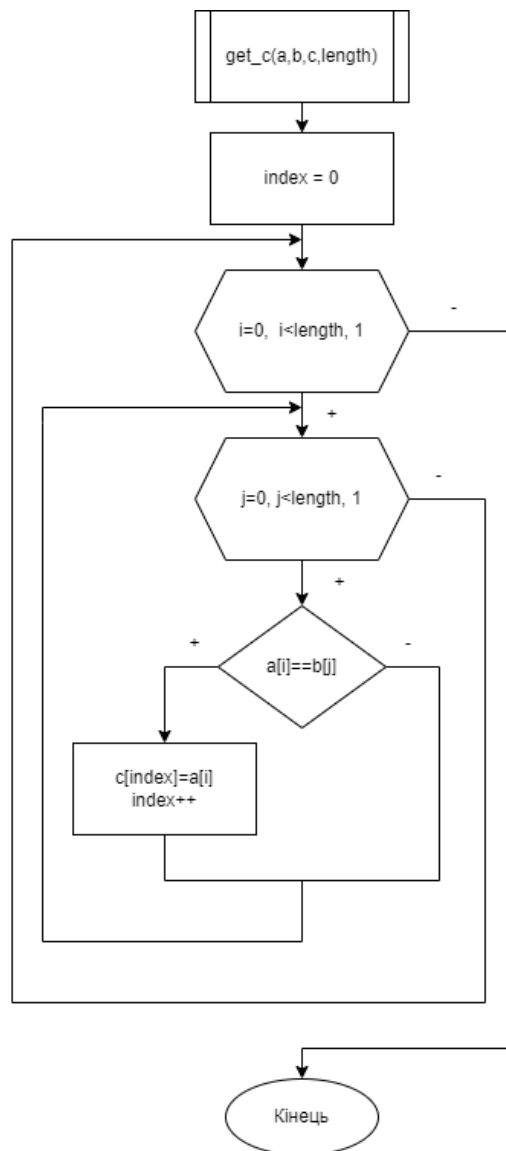
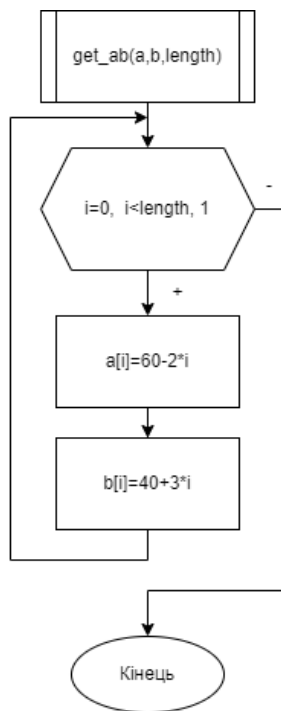
все повторити

Кінець

Блок-схема



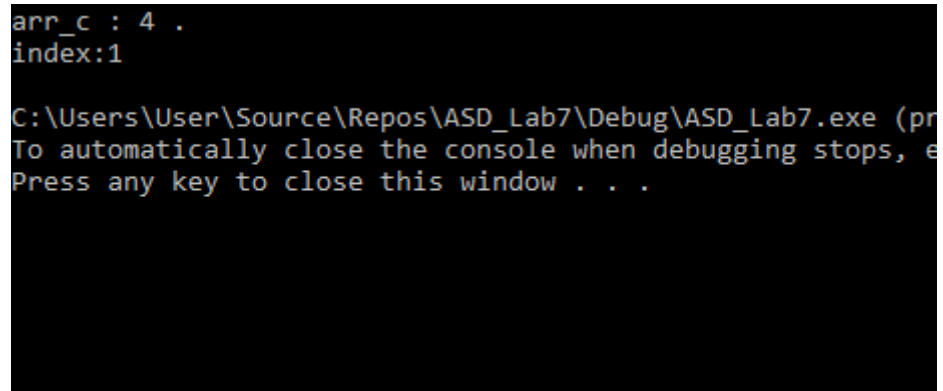
Підпрограми



Код

```
1  #include <iostream>
2  using namespace std;
3
4  void get_ab(char a[], char b[],int lenght) {
5      for (int i = 0; i < lenght; i++) {
6          a[i] = 60 - 2 * i;
7          b[i] = 40 + 3 * i;
8      }
9  }
10
11 void get_c(char a[], char b[], char c[], int lenght) {
12     int index = 0;
13     for (int i = 0; i < lenght; i++) {
14         for (int j = 0; j < lenght; j++) {
15             if (a[i] == b[j]) {
16                 c[index] = int(a[i]);
17                 index++;
18             }
19         }
20     }
21 }
22
23 int get_ind(char c[], int ind,int lenght) {
24     for (int i = 0; i < lenght; i++) {
25         if (c[i] == 52) {
26             ind = i;
27             i = lenght;
28         }
29     }
30     return ind;
31 }
32
33 int main(){
34     int ind = 0;
35     const int lenght = 10;
36     char a[lenght], b[lenght], c[lenght] = { 0 };
37     get_ab(a, b, lenght);
38     get_c(a, b, c, lenght);
39     ind = get_ind(c, ind, lenght);
40     cout << "arr[" << c << " ";
41     for (int i = 0; i < lenght; i++) {
42         cout << c[i] << " ";
43     }
44     cout << "" << endl;
45     cout << "index:" << ind <<endl;
46 }
```

Тестування



```
arr_c : 4 .  
index:1  
  
C:\Users\User\Source\Repos\ASD_Lab7\Debug\ASD_Lab7.exe (pr  
To automatically close the console when debugging stops, e  
Press any key to close this window . . .
```

Висновок

Висновок: Ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набули практичних навичок їх використання під час складання програмних специфікацій.