

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 № 6 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження лінійного пошуку в послідовностях»

Варіант 32

Виконав студент ІІ-12 Федій Олександр Валерійович

(шифр, прізвище, ім'я, по батькові)

Перевірів \_\_\_\_\_

( прізвище, ім'я, по батькові)

Київ 2021

## Лабораторна робота 7

### Дослідження лінійного пошуку в послідовностях

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

#### Варіант 32

**Задача 6.32.** 1. Описати три змінні індексованого типу з 10 символічних значень.

2. Ініціювати дві змінні виразами:

a.  $74 - i$

b.  $65 + 2 * i$

3. Ініціювати третю змінну рівними значеннями двох попередніх змінних.

4. Знайти кількість елементів, коди яких менше 67.

**Постановка задачі.** Результатом розв'язку є кількість елементів з кодами менше 67 масиву, що складається з однакових елементів двох масивів, значення яких задаються формулами із умови. Ввідних даних не вимагається.

**Математична побудова.** Складемо таблицю змінних.

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	A[10]	Проміжне дане
Другий масив	Символьний	B[10]	Проміжне дане
Масив спільних елементів	Символьний	C[10]	Проміжне дане
Кількість елементів у масиві C	Цілий	k	Проміжне дане
Кількість ел.	Цілий	n	Результат

<b>Масив</b>	Символьний	arr[]	Проміжне дане
<b>Доданок із заданої формули</b>	Цілий	add	Проміжне дане
<b>Множник із заданої формули</b>	Цілий	mult	Проміжне дане
<b>Лічильники</b>	Цілий	i, l	Проміжне дане

### ***Розв'язання:***

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

**Крок 1.** Визначимо основні дії.

**Крок 2.** Деталізуємо дію визначення перших двох масивів.

**Крок 3.** Деталізуємо дію визначення третього масиву.

**Крок 4.** Деталізуємо дію визначення кількості елементів менше 67.

### ***Псевдокод***

*крок 1*

***початок***

Знаходження першого та другого масиву

Знаходження третього масиву

Знаходження кількості елементів

***кінець***

*крок 2*

***початок***

A = arrayInit(A, 74, -1)

B = arrayInit(B, 65, 2)

знаходження третього масиву

знаходження кількості елементів

***кінець***

*крок 3*

***початок***

A = arrayInit(A, 74, -1)

B = arrayInit(B, 65, 2)

C, k = arrayEqual(A, B, C)

знаходження кількості елементів

***кінець***

*крок 4*

***початок***

A = arrayInit(A, 74, -1)

B = arrayInit(B, 65, 2)

C, k = arrayEqual(A, B, C)

n = arrayCheck(C, k)

***кінець***

***Псевдокод підпрограми***

***Функція*** arrayInit(arr[], add, mult):

**повторити** для i = 0; i < 10; i++

        arr[i] = add + mult \* i

**все повторити**

    arrayOutput(arr, 10)

**повернути** arr

***Кінець функції***

***Функція*** arrayOutput(arr[], k):

**повторити** для i = 0; i < k; i++

**виведення** arr[i]

**все повторити**

***Кінець функції***

**Функція** arrayEqual(A[], B[], C[]):

k = 0

**повторити** для i = 0; i < k; i++

**повторити** для l = 0; l < k; l++

**якщо** A[i] = B[l]

C[k]=A[i]

k++

**все якщо**

**все повторити**

**все повторити**

arrayOutput(C, k)

**повернути** C, k

**Кінець функції**

**Функція** arrayCheck(C[], k):

n = 0

**повторити** для i = 0; i < k; i++

**якщо** C[i] < 67

n++

**виведення** C[i]

**все якщо**

**все повторити**

**повернути** n

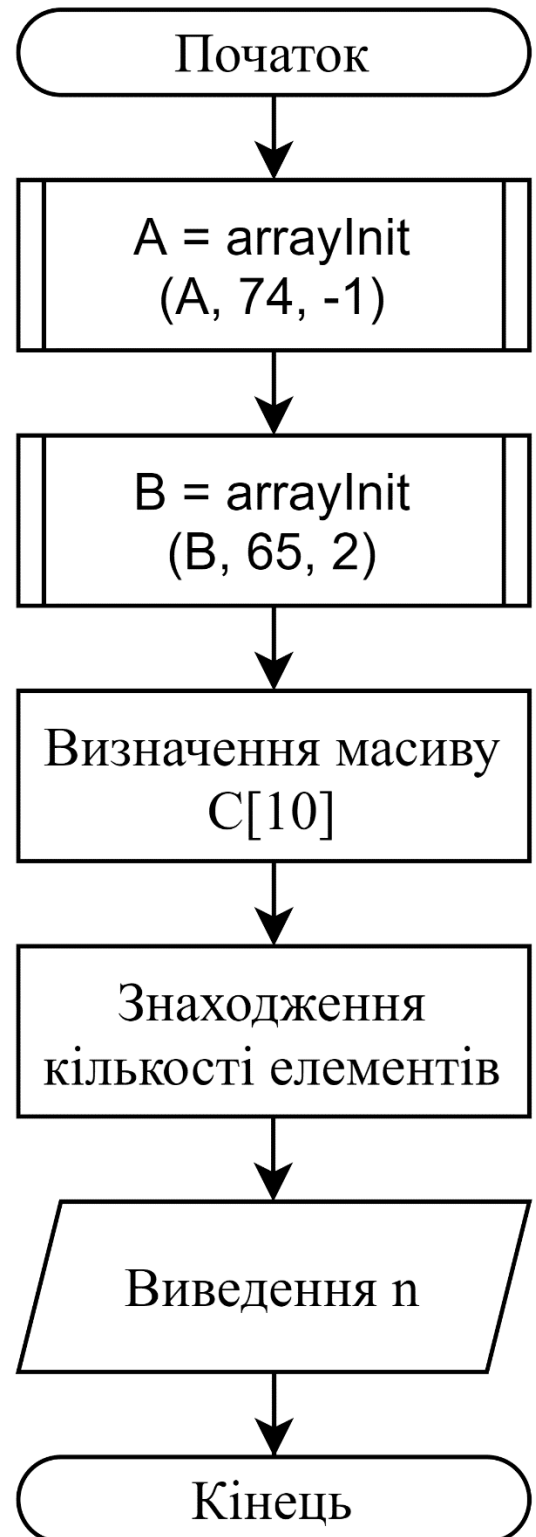
**Кінець функції**

*Блок-схема*

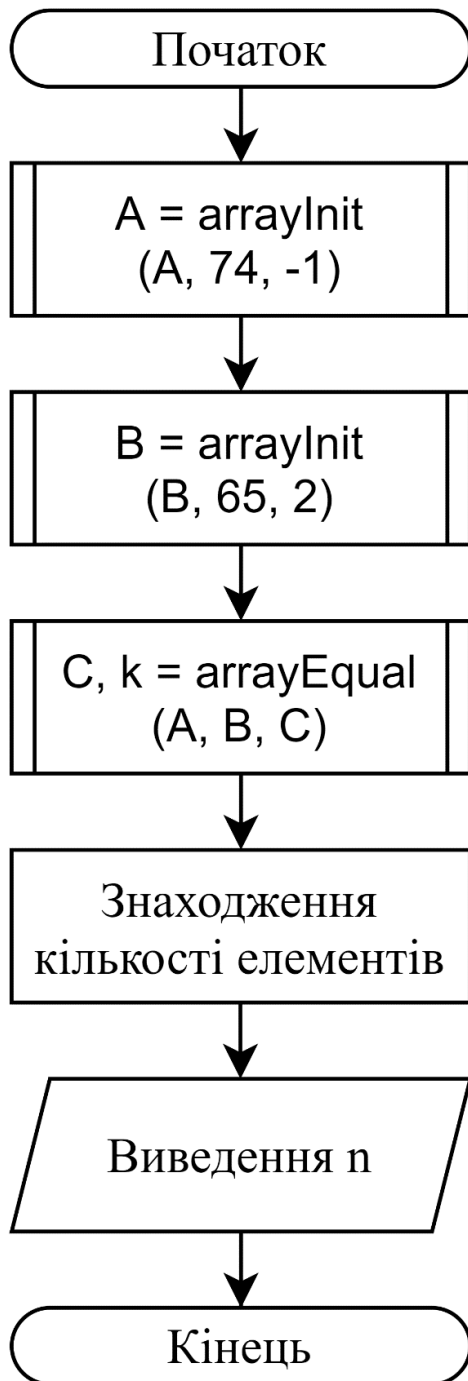
Крок 1



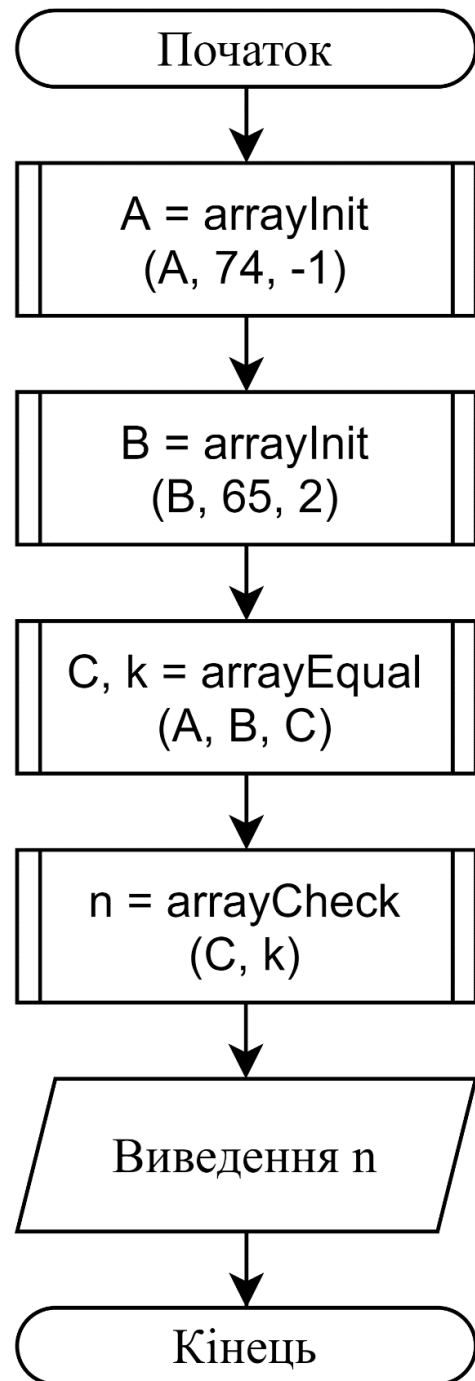
Крок 2

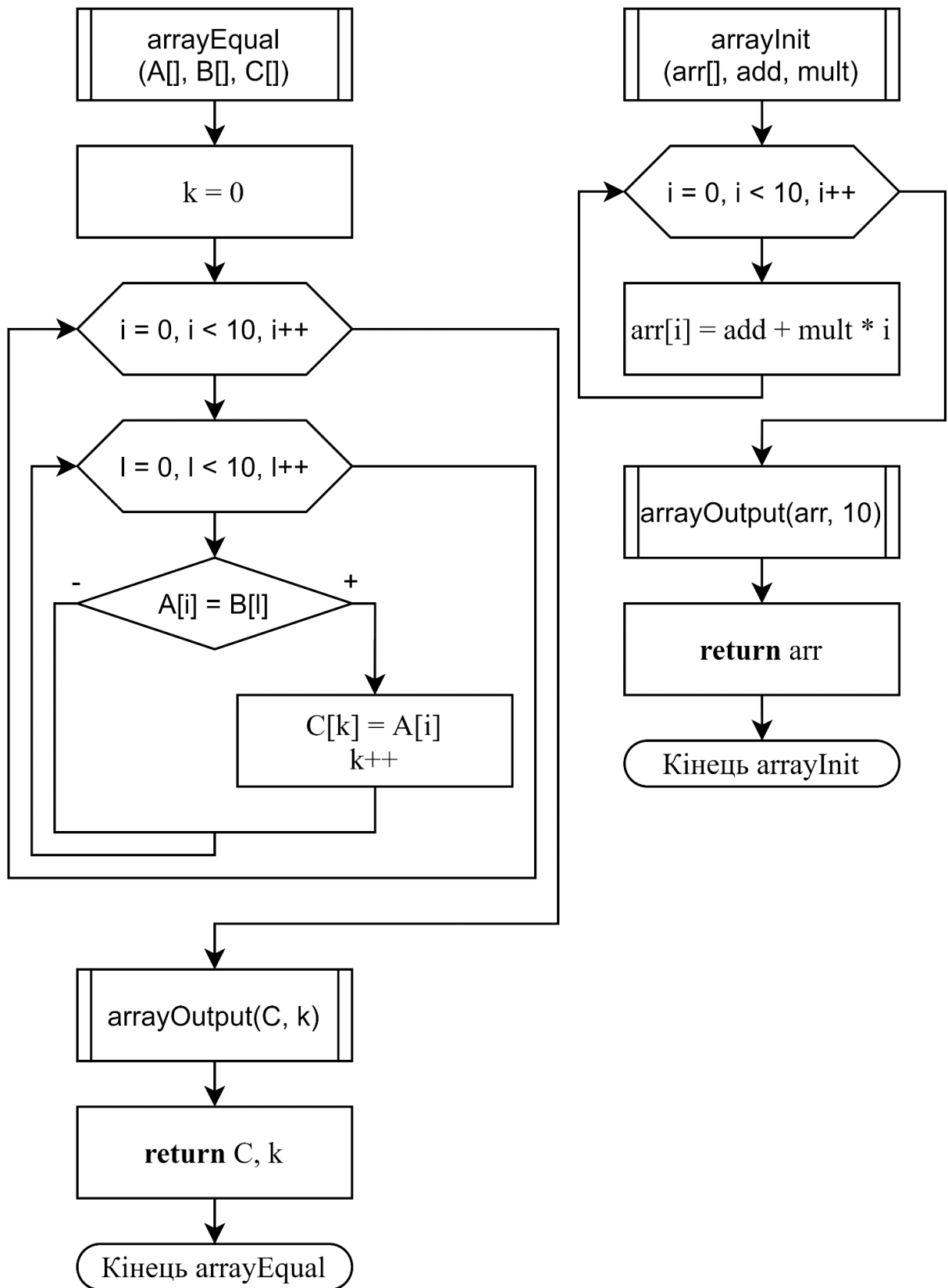


### Крок 3

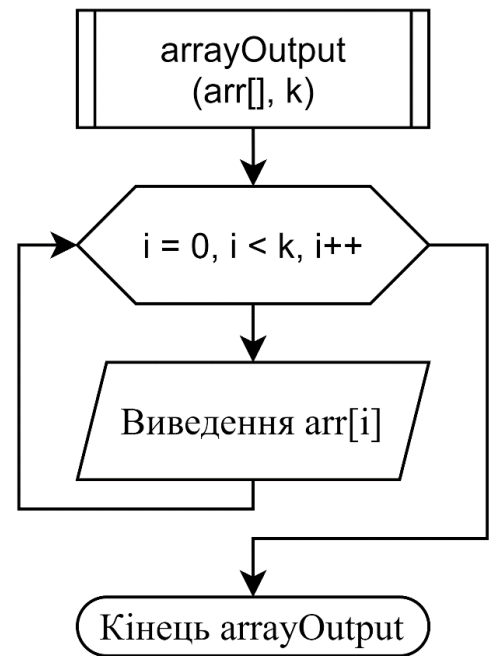
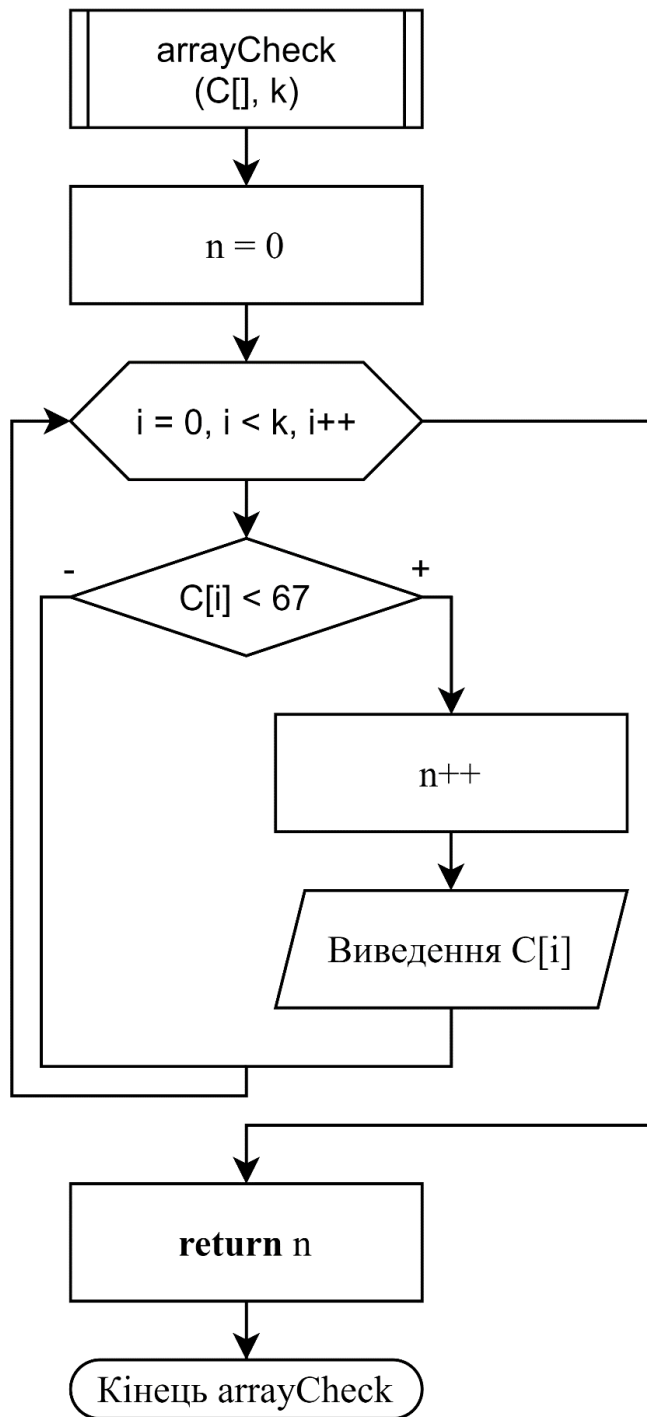


### Крок 4









### *Випробування алгоритму*

Блок	
	Початок
1	Масив А: $i = 0, A[0] = 74 - 0 = 74$ , ASCII – <b>J</b> Масив В: $i = 0, B[0] = 65 + 2 * 0 = 65$ , ASCII – <b>A</b>
2	Масив А: $i = 1, A[1] = 74 - 1 = 73$ , ASCII – <b>I</b> Масив В: $i = 1, B[0] = 65 + 2 * 1 = 67$ , ASCII – <b>C</b>
3	Масив А: $i = 2, A[2] = 74 - 2 = 72$ , ASCII – <b>H</b> $i = 2, B[0] = 65 + 2 * 2 = 69$ , ASCII – <b>E</b>
4	...
5	Масив А: $i = 9, A[9] = 74 - 9 = 65$ , ASCII – <b>A</b> Масив В: ...
	Вивід: 1
	Кінець

## Код програми

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  char* arrayInit(char arr[], int add, int mult);
6  int arrayEqual(char A[], char B[], char C[]);
7  int arrayCheck(char C[], int k);
8  void arrayOutput(char arr[], int k);
9
10 int main() {
11     //визначення першого масиву
12     char* A = new char[10];
13     cout << "A: ";
14     A = arrayInit(A, 74, -1);
15
16     //визначення другого масиву
17     char* B = new char[10];
18     cout << endl << "B: ";
19     B = arrayInit(B, 65, 2);
20
21     //визначення третього масиву
22     char C[10];
23     cout << endl << "C: ";
24     int k = arrayEqual(A, B, C);
25
26     //знаходження елементів менше 67
27     cout << endl << "Elements under 67: ";
28     int n = arrayCheck(C, k);
29     cout << endl << "n: " << setw(3) << n << endl;
30
31     system("pause");
32 }
```

```
34 char* arrayInit(char arr[], int add, int mult)
35 {
36     for (int i = 0; i < 10; i++)
37     {
38         arr[i] = add + mult * i;
39     }
40     arrayOutput(arr, 10);
41     return arr;
42 }
43
44 int arrayEqual(char A[], char B[], char C[])
45 {
46     int k = 0;
47     for (int i = 0; i < 10; i++)
48     {
49         for (int l = 0; l < 10; l++)
50         {
51             if (A[i] == B[l])
52             {
53                 C[k] = A[i];
54                 k++;
55             }
56         }
57     }
58     arrayOutput(C, k);
59     return k;
60 }
```

```

61  int arrayCheck(char c[], int k)
62  {
63      int n = 0;
64      for (int i = 0; i < k; i++)
65      {
66          if (c[i] < 67)
67          {
68              n++;
69              cout << setw(3) << c[i];
70          }
71      }
72      return n;
73  }
74
75  void arrayOutput(char arr[], int k)
76  {
77      for (int i = 0; i < k; i++)
78      {
79          cout << setw(3) << arr[i];
80      }
81  }

```

```

A:  J  I  H  G  F  E  D  C  B  A
B:  A  C  E  G  I  K  M  O  Q  S
C:  I  G  E  C  A
Elements under 67:  A
n:  1
Press any key to continue . . .

```

### ***Висновок***

Під час виконання лабораторної роботи було досліджено методи послідовного пошуку у впорядкованих і невлпорядкованих послідовностях та набуто практичних навичок їх використання під час складання програмних специфікацій. При виконанні лабораторної роботи було використано лінійний пошук – послідовний пошук даних, який виконується за допомогою оператора повторення з укладеним умовним оператором. Даний пошук використовувався над послідовностями значень або масивами, що

розглядається як іменована сукупність значень одного типу, а кожне значення має свій індекс.