

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 25

Виконав ПІ-15, Плугатирьов Дмитро Валерійович

студент (шифр, прізвище, ім'я, по батькові)

Перевірів Всечерковська Анастасія Сергіївна

(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 25

#### Завдання

25.) 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 (табл. 1).
3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
4. Обробки третьої змінної згідно з варіантом.

№	Вираз для обчислення елемента		Знайти
	1-го масиву	2-го масиву	
25	$100 + i$	$100 - i * i$	Добуток кодів елементів більших за 100

### 1. Постановка задачі

Описати три змінні індексованого типу з 10 символьних значень з таблиці ASCII. Ініціювати їх виразами з таблиці. Ініціювати третю змінну рівними значеннями першої та другої. Знайти добуток кодів елементів більших за 100 за допомогою вкладеного циклу та умови на величину значення символьних змінних.

## 2. Побудова математичної моделі

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Розмір масивів	Цілочисельний	SIZE	Початкові дані
Перший масив	Символьний	arr1	Початкові дані
Другий масив	Символьний	arr2	Початкові дані
Третій масив	Символьний	arr3	Проміжні дані
Добуток	Цілочисельний	multiplication	Результат
Індекс третього масиву	Цілочисельний	temp	Проміжні дані
Добуток елементів третього масиву	Підпрограма	mult_elements	Проміжні дані
Ініціалізація масивів	Підпрограма	init_arrs	Початкові дані
Лічильник циклу	Цілочисельний	i	Проміжні дані
Лічильник вкладеного циклу	Цілочисельний	z	Проміжні дані

- Реалізація пошуку рівних значень у першому та другому масивах відбувається шляхом проходження по елементам другого масиву з використанням вкладеного циклу і порівняння кожного з них із першим

першого з подальшим збільшенням його значення на одиницю. В разі рівності двох елементів їх значення заноситься до третього масиву.

Дія  $x++$  означає  $x := x + 1$ .

Дія  $x *= i$  означає  $x := x * i$ .

Дія  $(int)x$  означає приведення змінної  $x$  до цілочисельного типу.

### **3. Розв'язання**

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

*Крок 1.* Визначити основні дії.

*Крок 2.* Опис й ініціювання першого та другого масивів.

*Крок 3.* Ініціювання третього масиву, пошук добутку елементів третього масиву з індексом більше 100.

### **4. Псевдокод**

#### **Основна програма:**

*Крок 1*

**початок**

опис й ініціювання першого та другого масивів

ініціювання третього масиву, пошук добутку елементів третього масиву з індексом більше 100

**кінець**

*Крок 2*

**початок**

SIZE := 10

**init\_arrs(arr1, arr2, arr3, SIZE)**

ініціювання третього масиву, пошук добутку елементів третього масиву  
з індексом більше 100

**кінець**

*Крок 3*

**початок**

SIZE := 10

**init\_arrs(arr1, arr2, arr3, SIZE)**

**mult\_elements(arr3, SIZE)**

**кінець**

**Підпрограми:**

**init\_arrs(arr1, arr2, arr3, SIZE)**

temp := 0

**повторити для i від 0 до SIZE**

arr1[i] := 100 + i

arr2[i] := 110 - i \* i

**все повторити**

**повторити для i від 0 до SIZE**

**повторити для z від 0 до SIZE**

**якщо** arr1[i] == arr2[z]

**то**

arr3[temp] := arr1[i]

temp++

**все якщо**

**все повторити**

**все повторити**

**кінець**

**mult\_elements(arr3, SIZE)**

multiplication := 1

**повторити для i від 0 до SIZE**

**якщо** arr3[i] > 100

**то**

multiplication \*= arr3[i]

**все якщо**

**все повторити**

**якщо** multiplication == 1

**то**

multiplication := 0

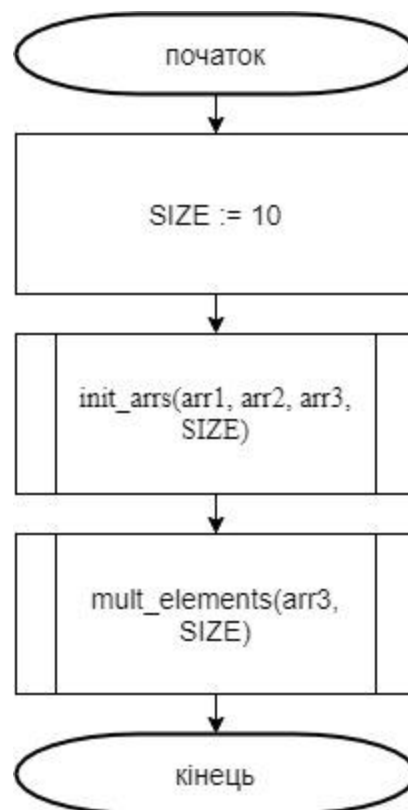
**все якщо**

**вивести** multiplication

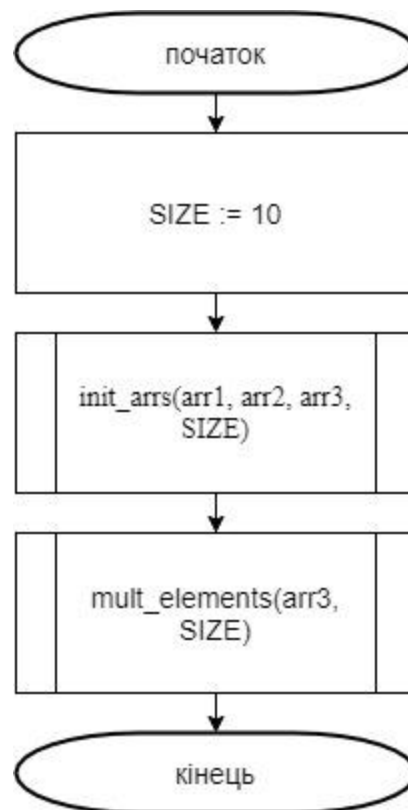
**кінець**

**Основна програма:**

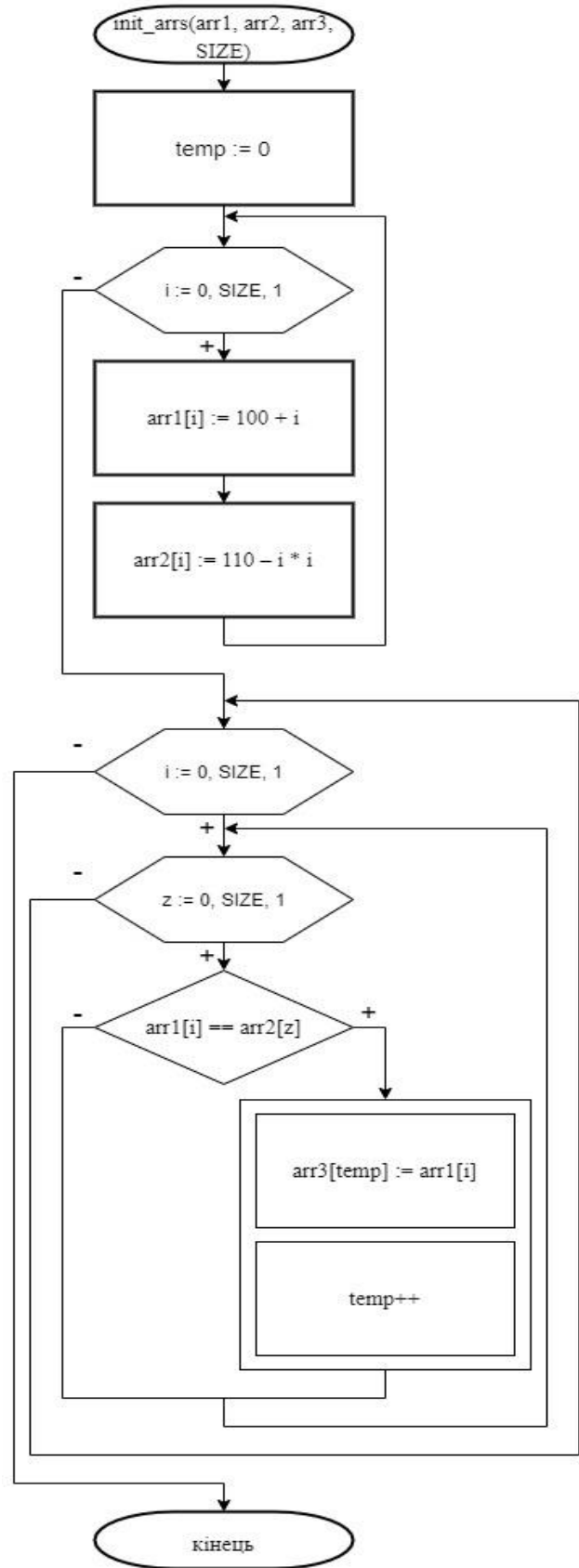
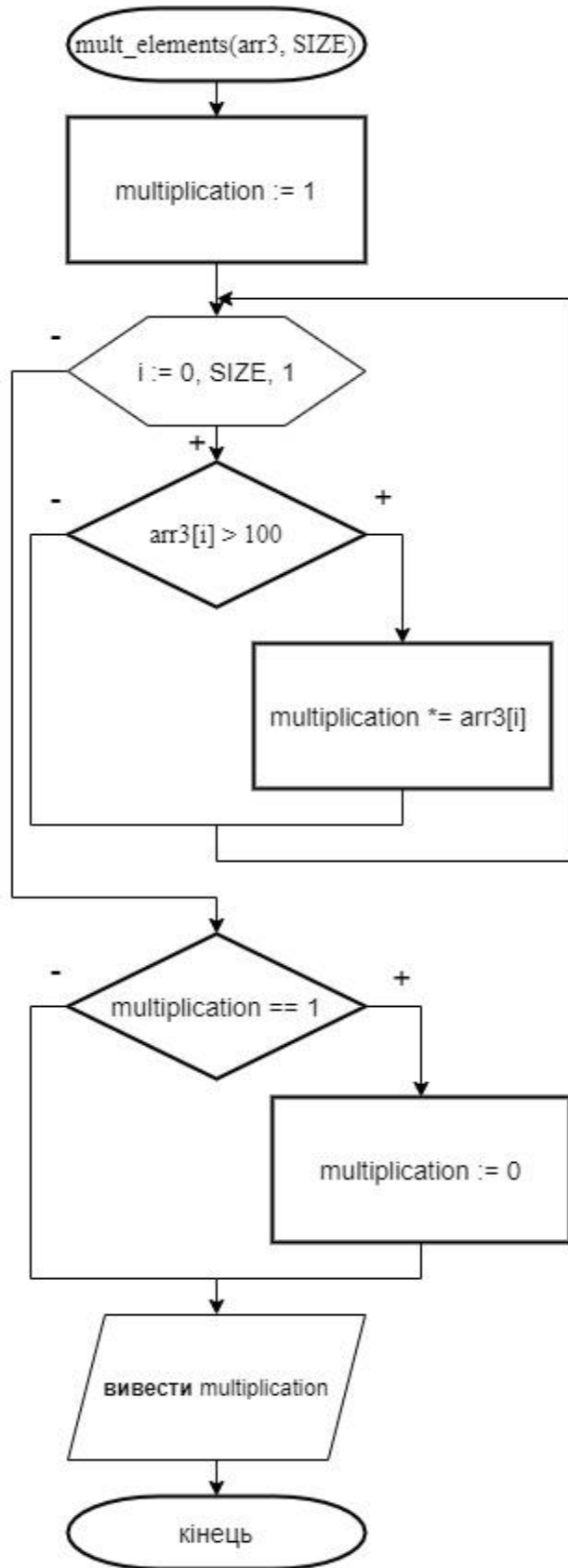
*Блок-схема*



## Підпрограми:







## 5. Код програми

```
main.cpp -# X
Lab. work 7 (Глобальная область)

1  #include <iostream>
2
3  // TASK 25
4  /*
5   * Розробити алгоритм та написати програму, яка складається з наступних дій:
6   * 1. Опису трьох змінних індексованого типу з 10 символних значень.
7   * 2. Ініціювання двох змінних виразами згідно з варіантом (табл. 1).
8   * 3. Ініціювання третьої змінної рівними значеннями двох попередніх змінних.
9   * 4. Обробки третьої змінної згідно з варіантом
10  * arr1 - 100 + i
11  * arr2 - 100 - i * i
12  * Добуток кодів елементів більших за 100
13  */
14
15  void output_arr(const char* arr, const int SIZE);
16  void mult_elements(char* arr3, const int SIZE);
17  void init_arrs(char* arr1, char* arr2, char *arr3, const int SIZE);
18
19  int main()
20  {
21      const int SIZE = 10;
22      char arr1[SIZE]{}; arr2[SIZE]{}; arr3[SIZE]{};
23      init_arrs(arr1, arr2, arr3, SIZE);
24      mult_elements(arr3, SIZE);
25  }
26
27  // Initialization of arrays
28  void init_arrs(char *arr1, char* arr2, char *arr3, const int SIZE)
29  {
30      int temp = 0;
31      for (int i = 0; i < SIZE; i++) // Initialization of the second and the first array
32      {
33          arr1[i] = 100 + i;
```

```
main.cpp -# X
Lab. work 7 (Глобальная область) init_arrs(char * arr1, char * arr2, char * arr3, const int SIZE)

34      arr2[i] = 110 - i * i;
35      }
36      output_arr(arr1, SIZE);
37      output_arr(arr2, SIZE);
38
39      for (int i = 0; i < SIZE; i++) // Initialization of the third array
40      {
41          for (int z = 0; z < SIZE; z++)
42          {
43              if (arr1[i] == arr2[z])
44              {
45                  arr3[temp] = arr1[i];
46                  temp++;
47              }
48          }
49      }
50      output_arr(arr3, SIZE);
51      std::cout << std::endl;
52  }
53
54  // Calculating of the multiplication of the elements of the third array
55  void mult_elements(char* arr3, const int SIZE)
56  {
57      int multiplication = 1;
58      for (int i = 0; i < SIZE; i++)
59      {
60          if (arr3[i] > 100)
61          {
62              multiplication *= arr3[i];
63              std::cout << "The element is: " << arr3[i] << " = " << (int)arr3[i] << std::endl << "The multiplication is: " << multiplication << std::endl;
64          }
65      }
66      (multiplication == 1) ? multiplication = 0 : multiplication;
```

```

main.cpp  X
Lab. work 7  (Глобальная область)  init_arrs(ch
64         }
65     }
66     (multiplication == 1) ? multiplication = 0 : multiplication;
67     std::cout << "The multiplication of elements whose code is bigger than 100 is: " << multiplication << std::endl;
68 }
69
70 // The output of an array
71 void output_arr(const char* arr, const int SIZE)
72 {
73     std::cout << "Here is an array: " << std::endl;
74     for (int i = 0; i < SIZE; i++)
75     {
76         std::cout << arr[i] << "(" << (int)arr[i] << ")" << " ";
77     }
78     std::cout << std::endl;
79 }

```

## 5. Т е с т у в а н н я

Консоль отладки Microsoft Visual Studio

```

Here is the 1 array:
d(100) e(101) f(102) g(103) h(104) i(105) j(106) k(107) l(108) m(109)
Here is the 2 array:
n(110) m(109) j(106) e(101) ^ (94) U(85) J(74) =(61) .(46) ⇨(29)
Here is the 3 array:
e(101) j(106) m(109) (0) (0) (0) (0) (0) (0) (0) (0)

The element is: e = 101
The multiplication is: 101
The element is: j = 106
The multiplication is: 10706
The element is: m = 109
The multiplication is: 1166954
The multiplication of elements whose code is bigger than 100 is: 1166954

```

Блок	Дія 1
	Початок
1	SIZE := 10
2	arr1[SIZE], arr2[SIZE], arr3[SIZE]
3	init_arrs(arr1, arr2, arr3, SIZE)
4	temp := 0
5	i := 0
6	arr1[i] := 100 + i
7	arr2[i] := 110 - i * i
...	...
166	arr3[temp] := arr1[i]
167	temp++
...	...
509	multiplication *= arr3[i]
...	...
538	Вивести multiplication

## **6. В и с н о в о к**

В цій лабораторній роботі я дослідив методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набув практичних навичок їх використання під час складання програмних специфікацій. Мені довелося ініціювати три символьні масиви та працювати з їх елементами, вважаючи їх за числа або символи в залежності від конкретної ситуації та порівнювати їх, використовуючи певні методи пошуку.