# 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 лінійних алгоритмів»

Варіант 25

Виконав студент ІП-11 Прищепа Владислав Станіславович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

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

Варіант 25

**Індивідуальне завдання:**

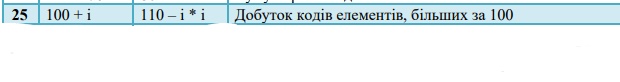
Розробити алгоритм та написати програму, яка складається з наступних дій:

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.



**Постановка задачі:**

Вивести 2 масиви з 10 символьних значень, коди яких дані в умові. Вивести 3 масив з елементів, яка складається з спільних елементів 1 та 2 масивів. Вивести добуток кодів елементів 3 масиву, коди яких більше 100.

**Математична модель:**

|  |  |  |  |
| --- | --- | --- | --- |
| Зміна | Тип | Ім’я | Призначення |
| Розмір масивів | const int | z | Константа |
| Масив 1 | char | arr1 | Результат |
| Масив 2 | char | arr2 | Результат |
| Масив 3 | char | arr3 | Результат |
| Розмір масива 3 | int | q | Проміжне знач. |
| Добуток кодів елементів масива 3 | int | dobutok | Результат |
| 1 елемент масиву 1 | char\* | p1 | Проміжне знач. |
| 1 елемент масиву 2 | char\* | p2 | Проміжне знач. |
| 1 елемент масиву 3 | char\* | p3 | Проміжне знач. |
| 1 елемент масиву | char\* | p | Проміжне знач. |
| значення розміру масива | int | v | Проміжне знач. |
| значення розміру масива 3 | int | t | Проміжне знач. |
| іттерація 1 | int | i | Проміжне знач. |
| іттерація 2 | int | k | Проміжне знач. |
| значення добутків масива | int | d | Проміжне знач. |

Заповнивши масиви 1 і 2 відповідними символьними значеннями, знайдемо їх спільні елементи лінійним пошуком та перенесемо їх у третій масив. Коди елементів 3-го масиву, які > 100, перемножимо і виведемо на екран значення.

**Розв’язання**

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

Крок 1: Визначимо основні дії

Крок 2: деталізуємо заповнення arr1 та arr2

Крок 3: деталізуємо знаходження спільних елементів arr1, arr2 та заповнимо 3-ій масив

Крок 4: деталізуємо виведення масивів

Крок 5: деталізуємо знаходження добутку кодів, більших 100

**Псевдокод**

**Основна програма**

**Початок**

z:=10

fill\_arr(arr1, arr2, z)

q = fill\_arr3(arr1, arr2, arr3, z)

output(arr1, z)

output(arr2, z)

output(arr3, q)

dobutok = code\_multiply(arr3, q)

**Виведення** dobutok

**Кінець**

**Підпрограма**

**fill\_arr**(p1, p2, v)

**для** і **від** 0 **до** v, **з кроком** 1 **повторити**

p1[i] = char(100 + i)

p2[i] = char(110 - i \* i)

**все повторити**

**Кінець**

**fill\_arr3**(p1, p2, p3, v)

t:=0

**для** і **від** 0 **до** v, **з кроком** 1 **повторити**

**для** k **від** 0 **до** v, **з кроком** 1 **повторити**

**Якщо** p1[i] == p2[k]

p3[t] := p1[i]

t := t+1

**Все якщо**

**все повторити**

**все повторити**

**return** t

**Кінець**

**output**(p, v)

**для** і **від** 0 **до** v, **з кроком** 1 **повторити**

**Виведення** p[i]

**все повторити**

**Виведення** endl

**Кінець**

**code\_multiply**(p, v)

d:=1

**для** і **від** 0 **до** v, **з кроком** 1 **повторити**

**Якщо** int(p[i]) > 100

d := d\*int(p[i])

**Все якщо**

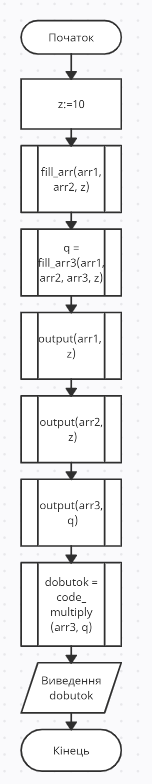
**все повторити**

**return** d

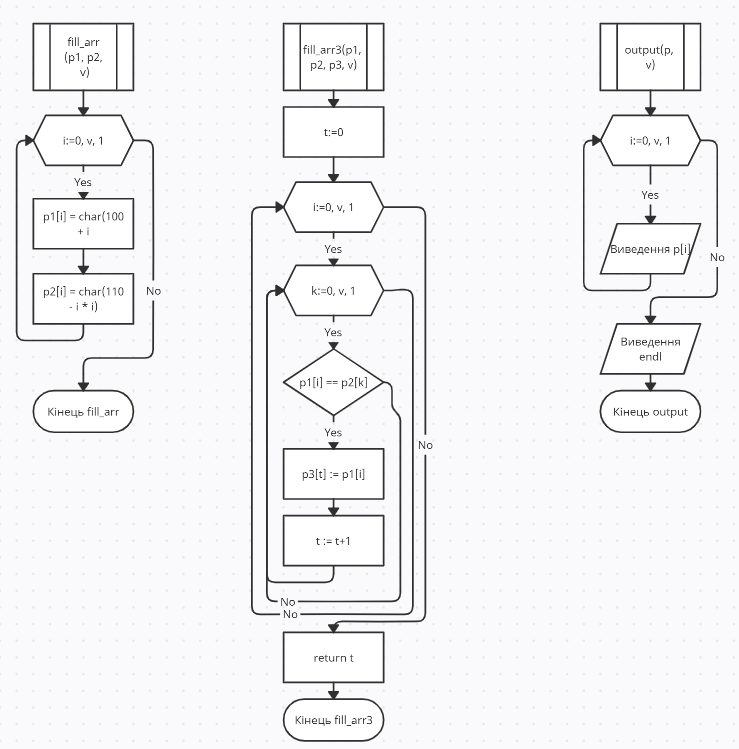
**Кінець**

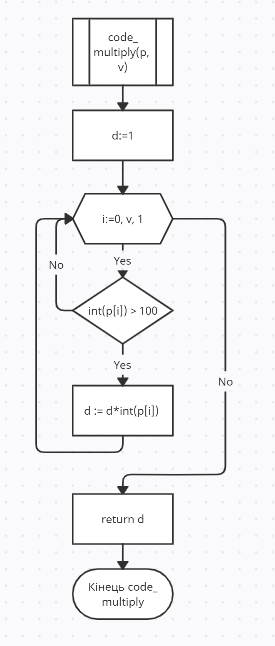
**Блок-схема алгоритму**

**Основна програма**

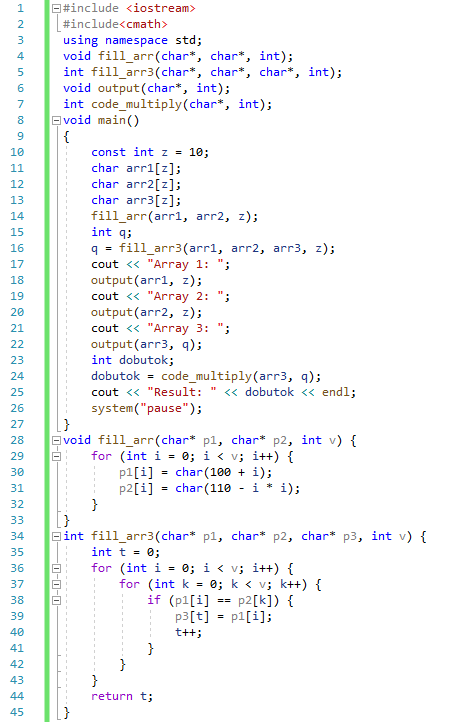


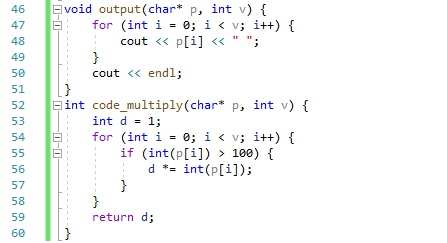
**Підпрограми**

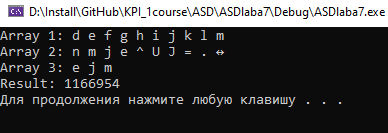




**Код та виконання коду на С++:**







**Випробування алгоритму**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер елемента масиву | Код 1 масиву | Елемент 1 масиву | Код 2 масиву | Елемент 1 масиву | Спільні коди (3 масив) | Елементи  спільні (3 масив) |
| 0 | 100 | d | 110 | n | 101 | e |
| 1 | 101 | e | 109 | m | 106 | j |
| 2 | 102 | f | 106 | j | 109 | m |
| 3 | 103 | g | 101 | e |  |  |
| 4 | 104 | h | 96 | ^ |  |  |
| 5 | 105 | i | 85 | U |  |  |
| 6 | 106 | j | 84 | J |  |  |
| 7 | 107 | k | 61 | = |  |  |
| 8 | 108 | l | 46 | . |  |  |
| 9 | 109 | m | 29 | ↔ |  |  |

Всі коди 3 масиву більше 100, їх добуток = 1166954.

**Висновок**

Отже, у результаті лабораторної роботи ми дослідили особливості методів лінійного пошуку в послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм; навчилися будувати математичну модель, що відповідає постановці задачі, псевдокод та блок-схеми, які пояснюють логіку алгоритму. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження добутку спільних кодів елементів, більших за 100, декомпозували задачу на 5 кроків: визначили основні дії, деталізували заповнення arr1 та arr2, знаходження спільних елементів arr1, arr2 та заповнимо 3-ій масив, виведення масивів та знаходження добутку кодів, більших 100. У алгоритмі використовується підпрограма лінійного пошуку спільних елементів 1 та 2 масивів, які вводяться у 3 масив. Оскільки формула перевірені вручну результати розрахунку членів послідовності відповідають дійсності, то алгоритм правильно подає результат.