

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 10

Виконав студент ІП-11, Друзенко Олександра Юріївна
(шифр, прізвище, ім'я, по батькові)

Перевірив Мартінова Оксана Петрівна
(прізвище, ім'я, по батькові)

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 10

10	$100 + i$	$110 - i * i$	Добуток кодів елементів
----	-----------	---------------	-------------------------

1. Постановка задачі

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

1. Опису трьох змінних індексованого типу з 10 символічних 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.

2. Математична модель

Змінна	Тип	Ім'я	Призначення
Перший масив	array of char	array1	проміжне дане
Другий масив	array of char	array2	проміжне дане
Третій масив	array of char	array3	проміжне дане
Добуток кодів символів	int	res	результат
Показчик заповнення	bool	full	проміжне дане
Функція виведення масиву	function	outputArray	функція виведення
Функція ініціалізації масивів 1 і 2	function	initArray	функція ініціалізації
Функція ініціалізації масиву 3 і знаходження результату	function	initThirdArrayFindRes	функція ініціалізації та обчислення

Функція print() – виведення на екран

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо підпрограму ініціалізації 1-го і 2-го масивів

Крок 3. Деталізуємо підпрограму ініціалізації 3-го масиву та знаходження результату

Крок 4. Деталізуємо підпрограму виведення масиву в консоль

3.Псевдокод

Крок 1. Основна програма

Початок

```
array1[10], array2[10], array3[10], full, res
initArray(array1, array2)
res = initThirdArrayFindRes(array1, array2, array3, full)
print ("first array: ")
outputArray(array1);
print ("second array: ")
outputArray(array2)
print ("third array: ")
outputArray(array3)
print ("result", res)
```

Кінець

Крок 2. Підпрограма ініціалізації 1-го і 2-го масивів

Початок

Функція *initArray (array1, array2):*

Повторити для i від 0 до 10

$array1[i] = 100 + i$

$array2[i] = 110 - i * i$

Все повторити

Кінець

Крок 3. Підпрограма ініціалізації 3-го масиву та знаходження результату

Початок

Функція *initThirdArrayFindRes(array1, array2, array3, full):*

$res = 1$

Повторити для i від 0 до 10

$full = 0$

Повторити для j від 0 до 10

Якщо $(array1[i] == array2[j])$

$array3[i] = array2[j];$

$full = 1;$

$res *= array3[i];$

Все якщо

Все повторити

Якщо $(full = 0)$

$array3[i] = 48;$

Все якщо

Все повторити

Повернути res

Кінець

Крок 4. Підпрограма виведення масиву в консоль

Початок

Функція outputArray(char array):

Повторити для i від 0 до 10

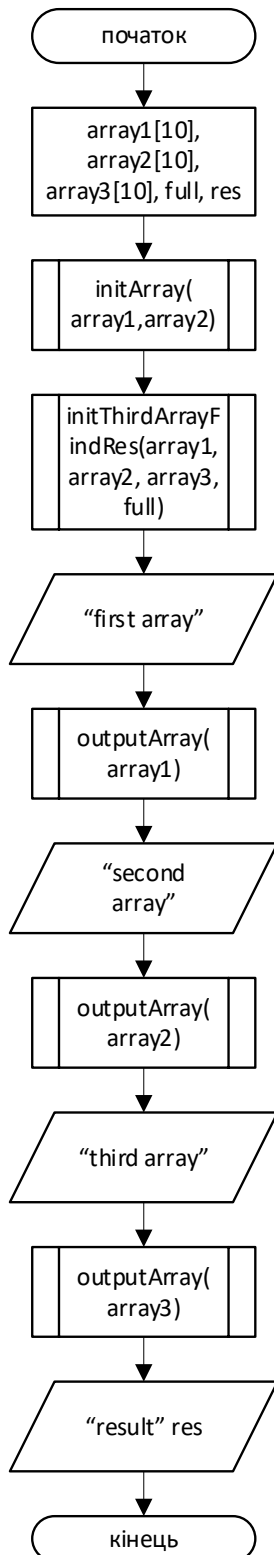
print(array[i])

Все повторити

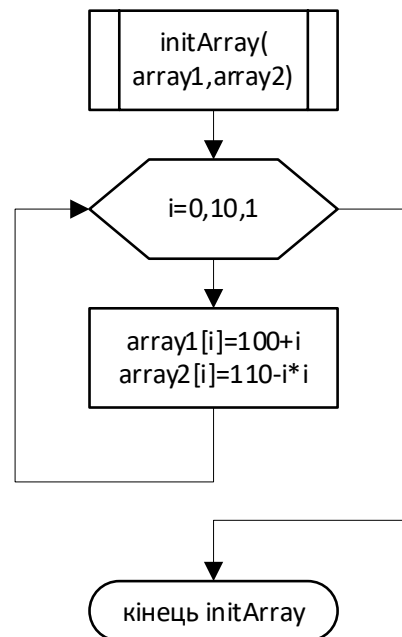
Кінець

4.Блок-схема

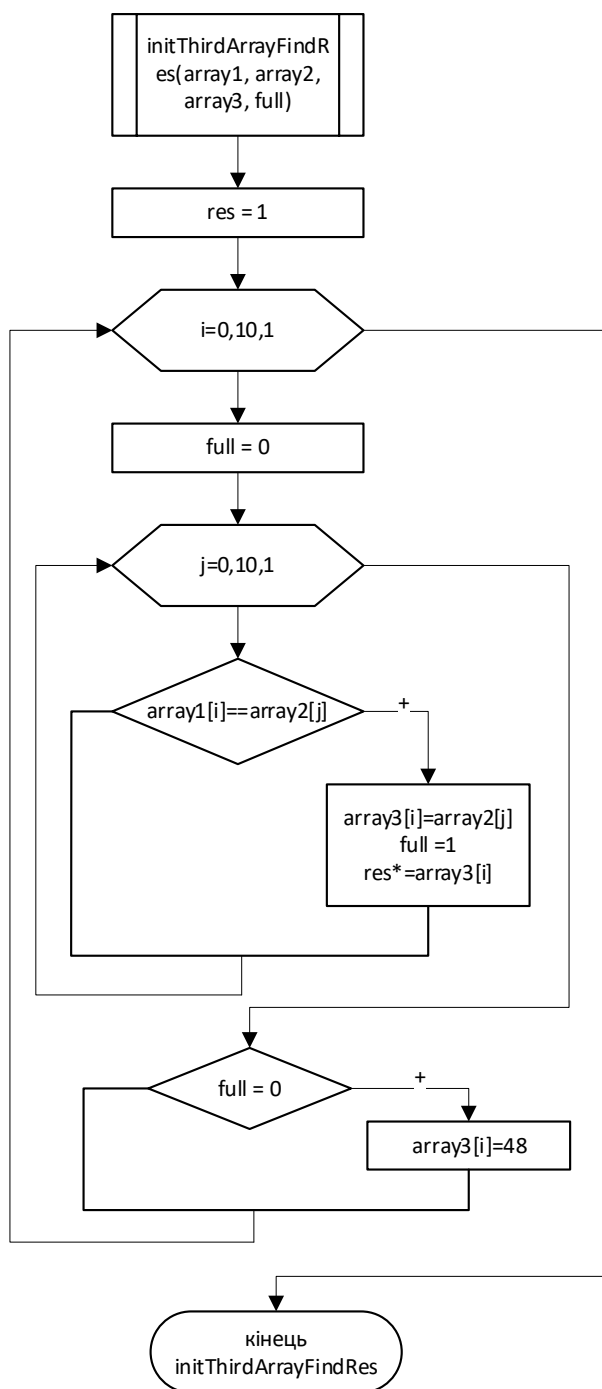
Крок 1



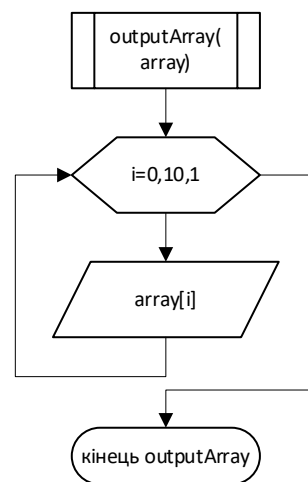
Крок 2. Функція ініціалізації 1-го і 2-го масивів



Крок 3. Функція ініціалізації 3-го масиву та знаходження результату



Крок 4. Функція виведення масиву в консоль



5. Код програми (C++)

```
#include <iostream>
using namespace std;

void outputArray(char array[]);
void initArray(char array1[], char array2[]);
int initThirdArray(char array1[], char array2[], char array3[], bool full);

int main()
{
    char array1[10];
    char array2[10];
    char array3[10];

    int res;
    bool full = 0;

    initArray(array1, array2);
    res = initThirdArray(array1, array2, array3, full);
    cout << "first array: "; outputArray(array1); cout << endl;
    cout << "second array: "; outputArray(array2); cout << endl;
    cout << "third array: "; outputArray(array3); cout << endl;
    cout << "result:" << res;

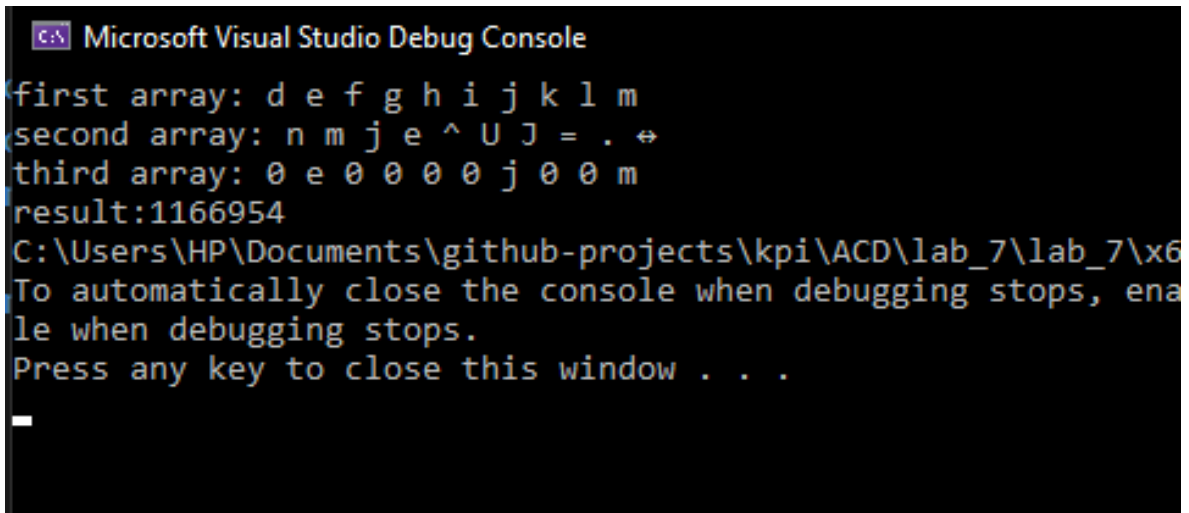
}

void outputArray(char array[]) {
    for (int i = 0; i < 10; i++) {
        cout << array[i]<< " ";
    }
}

void initArray(char array1[],char array2[]) {
    for (int i = 0; i < 10; i++) {
        array1[i] = 100 + i;
        array2[i] = 110 - i * i;
    }
};

int initThirdArray(char array1[], char array2[], char array3[], bool full) {
    int res = 1;
    for (int i = 0; i < 10; i++) {
        full = 0;
        for (int j = 0; j < 10; j++) {
            if (array1[i] == array2[j]) {
                array3[i] = array2[j];
                full = 1;
                res *= array3[i];
            }
        }
        if (!full) {
            array3[i] = 48;
        }
    }
    return res;
};
```

6. Тестування програми



```
Microsoft Visual Studio Debug Console
first array: d e f g h i j k l m
second array: n m j e ^ U J = . ↔
third array: 0 e 0 0 0 0 j 0 0 m
result:1166954
C:\Users\HP\Documents\github-projects\kpi\ACD\lab_7\lab_7\x6
To automatically close the console when debugging stops, ena
le when debugging stops.
Press any key to close this window . . .
```

крок	дія
	початок
1	first array: d e f g h i j k l m
2	second array: n m j e ^ U J = . ↔
3	third array: 0 e 0 0 0 0 j 0 0 m
4	res = 1166954 (e*j*m=101*106*109)
5	result: 1166954
	кінець

7. Висновок

Отже, сьогодні я дослідила методи послідовного пошуку у послідовностях та набула практичних навичок їх використання.

В результаті лабораторної роботи я розробила алгоритм який заповнює два масиви, знаходить спільні елементи та записує їх в третій масив, після чого обчислюється добуток цих елементів.

Використовуючи арифметичні цикли та розгалуження я зробила послідовний пошук у впорядкованих послідовностях.

Випробувавши алгоритм, я отримала шукані результати. Алгоритм працює.