

Додаток 1

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

«Дослідження лінійного пошуку в
послідовностях»

Варіант 4

Виконав студент Бутов Даниїл Романович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(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 4

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.
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	
4	$2 * i + 23$	$49 - 2 * i$	Різницю між кодами максимального та мінімального елементів

Постановка задачі.

Нам потрібно згенерувати два масиви за формулами з таблиці. Також створити третій масив який відповідає рівним значенням з першого та другого. Також результатом завдання буде виведення різниці між кодами максимального та мінімального елементів.

Побудова математичної моделі.

Змінна	Тип	Ім'я	Призначення
Перший масив	Символьний	first_array	Початкове
Другий масив	Символьний	second_array	Початкове
Третій масив	Символьний	third_array	Результат
Найбільший елемент	Символьний	max	Проміжні дані
Найменший елемент	Символьний	min	Проміжні дані
Генерування масивів a,b	Процедура	create_arrays()	Проміжні дані
Генерування масиву c	Процедура	find_array()	Проміжні дані
Пошук різниці між кодами	Процедура	ext()	Результат
Вивід масивів	Процедура	our_arr()	Результат

First_array та second_array згенуємо за формулами з таблиці. Third_array формуємо на основі перших двох масивів за допомогою for(), if(). Для знаходження різниці між максимальним та мінімальним значенням використовуємо вкладені цикли. Спочатку знайдемо max, min та віднімемо їх значення. Вивід масивів також зробимо через цикл for(). В тілі основної функції будуть лише виклики підпрограми та створення змінних масивів.

Розв'язання.

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

Крок 1. Визначемо основні дії.

Крок 2. Деталізуємо генерування масивів *first_array*, *second_array*.

Крок 3. Деталізуємо генерування масиву *third_array*.

Крок 4. Деталізуємо знаходження різниці між максимальним та мінімальним значенням.

Крок 5. Деталізуємо вивід масивів.

Псевдокод.

Крок 1

Початок

Генерування масивів *first_array*, *second_array*

Генерування масиву *third_array*.

Знаходження різниці між максимальним та мінімальним значенням.

Деталізуємо вивід масивів.

Кінець

Крок 2

Початок

create_arrays(first_array, second_array)

Генерування масиву *third_array*.

Знаходження різниці між максимальним та мінімальним значенням.

Деталізуємо вивід масивів.

Кінець

Крок 3

Початок

create_arrays(first_array, second_array)

find_array(first_array, second_array, third_array)

Знаходження різниці між максимальним та мінімальним значенням.

Деталізуємо вивід масивів.

Кінець

Крок 4

Початок

```
create_arrays(first_array, second_array)
find_array(first_array, second_array, third_array)
ext(third_array)
```

Деталізуємо вивід масивів.

Кінець

Крок 5

Початок

```
create_arrays(first_array, second_array)
find_array(first_array, second_array, third_array)
ext(third_array)
out_arr(first_arr, second_arr, third_arr)
```

Кінець

Підпрограми:

```
create_arrays(first , second )
```

Початок

```
для i від 0 до 10 повторити
    first[i] = 2 * i + 23
    second[i] = 49 - 2 * i
    якщо first[i] <= 32
        то
            first[i] = '0'
    все якщо
    якщо second[i] <= 32
        то
            second[i] = '0'
    все якщо
все повторити
```

Кінець

find_array(first, second, third)

Початок

```
    для i від 0 до 10 повторити
        для j від 0 до 10 повторити
            якщо first[i] == second[j]
                то
                    third[i] = first[i]
            все якщо
        все повторити
    все повторити
```

Кінець

ext(third)

Початок

```
    max = 33
    min = 33
    для i від 0 до 10 повторити
        якщо max < third[i] && third[i] != '0'
            то
                max = third[i]
        все якщо
        якщо min > third[i] && third[i] != '0'
            то
                min = third[i]
        все якщо
    все повторити
    Виведення max - min
```

Кінець

out_arr(first, second, third)

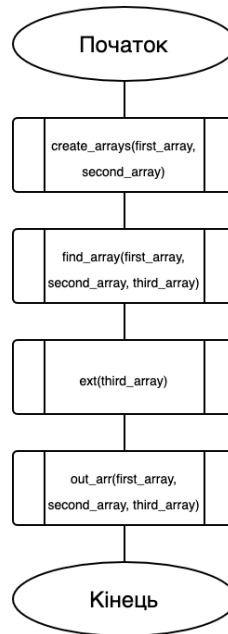
Початок

```
    для i від 0 до 10 повторити
        Виведення first[i], second[i], third[i]
    все повторити
```

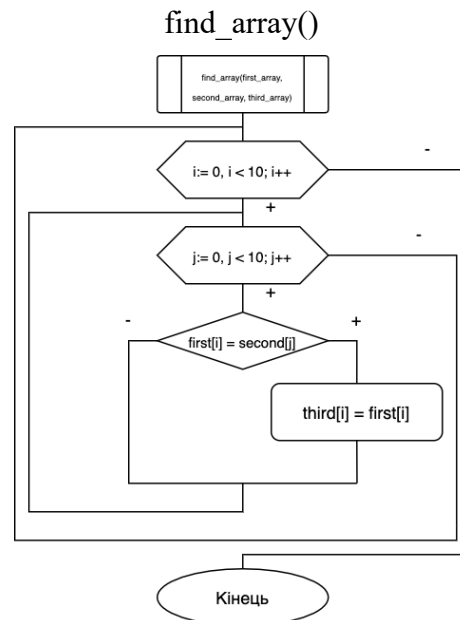
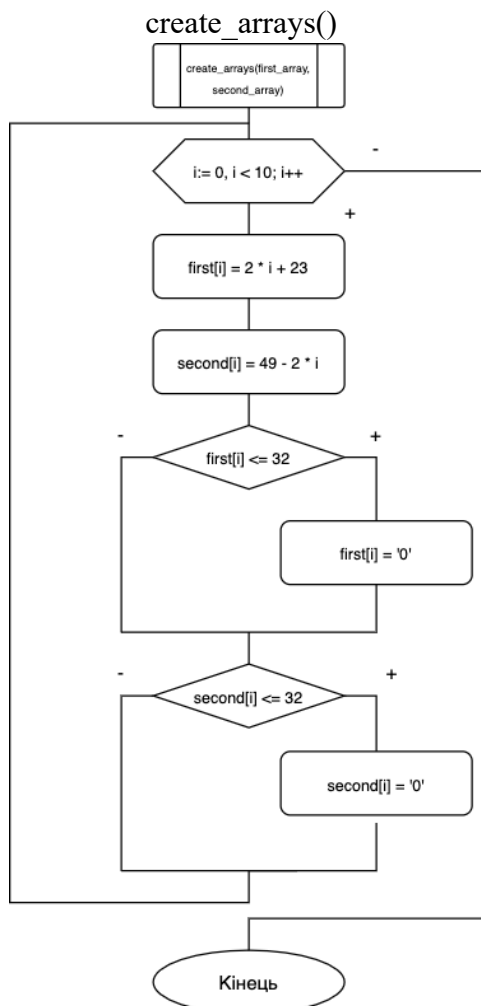
Кінець

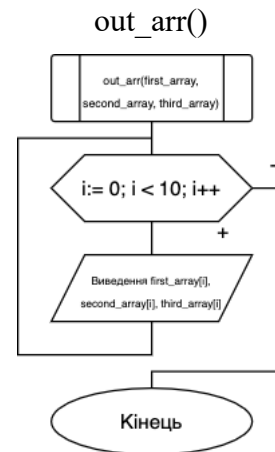
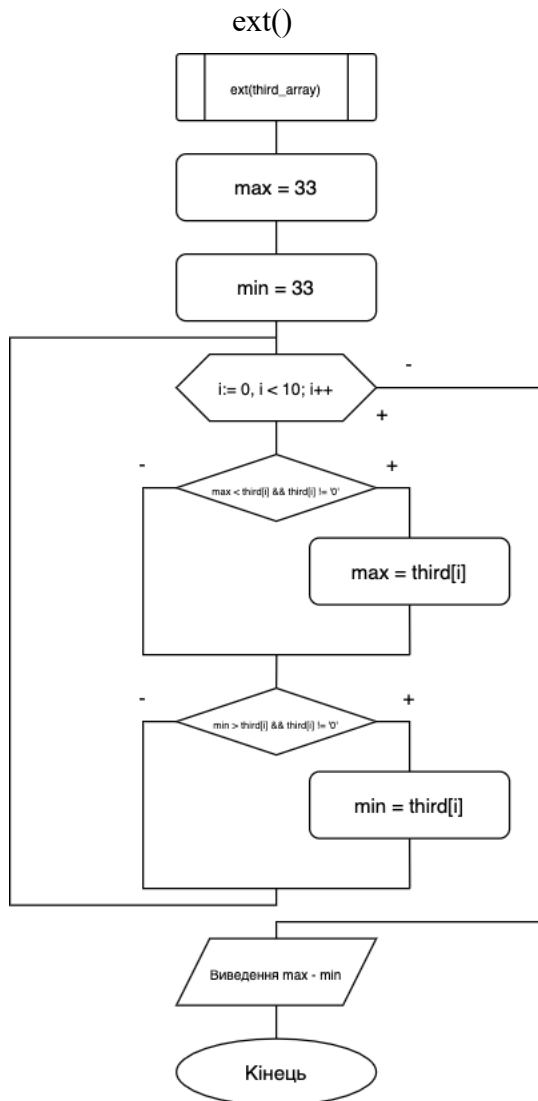
Блоксхема.

Основна програма.



Підпрограми.





Код.

```
1  #include <iostream>
2  using namespace std;
3
4  void create_arrays(char *first, char *second); // Створення масивів
5  void find_array(char *first, char *second, char *third); // Створення фінального масиву
6  void ext(char *third); // Пошук максимального значення
7  void out_arr(char *first, char *second, char *third); // Вивід масиву
8
9  int main() {
10     char first_arr[10], second_arr[10], third_arr[10];
11     create_arrays(first: first_arr, second: second_arr);
12     find_array(first: first_arr, second: second_arr, third: third_arr);
13     ext(third: third_arr);
14     out_arr(first: first_arr, second: second_arr, third: third_arr);
15 }
16 // Створення масивів
17 void create_arrays(char *first, char *second){
18     for (int i = 0; i < 10; ++i) {
19         first[i] = 2 * i + 23;
20         second[i] = 49 - 2 * i;
21         if (first[i] <= 32) first[i] = '0';
22         if (second[i] <= 32) second[i] = '0';
23     }
24 }
25 // Створення фінального масиву
26 void find_array(char *first, char *second, char *third){
27     for (int i = 0; i < 10; ++i) {
28         for (int j = 0; j < 10; ++j) {
29             if (first[i] == second[j]) third[i] = first[i];
30         }
31     }
32 }
33 // Пошук максимального значення
34 void ext(char *third) {
35     int max = 33, min = 33;
36     for (int i = 0; i < 10; ++i) {
37         if (max < third[i] && third[i] != '0') max = third[i];
38         if (min > third[i] && third[i] != '0') min = third[i];
39     }
40     cout << "Різниця між кодами елементів: " << max - min << endl;
41 }
42 // Вивід масиву
43 void out_arr(char *first, char *second, char *third){
44     for (int i = 0; i < 10; ++i) {
45         cout << first[i] << " " << second[i] << " " << third[i] << endl;
46     }
47 }
```

Тестування.

```
Різниця між кодами елементів: 8
0 1 0
0 / 0
0 - 0
0 + 0
0 ) 0
! ' !
# % #
% # %
' ! '
) 0 )
```

Process finished with exit code 0

Випробування.

Блок	Дія
	Початок
1	create_arrays(first_array, second_array)
	first_array = {0, 0, 0, 0, 0, !, #, %, '<')}
	second_array = {1, /, -, +,), ', %. #, !, 0}
	{23, 25, 27, 29, 31, 33, 35, 37, 39, 41}
	{49, 47, 45, 43, 41, 39, 35, 33, 31, 29}
2	find_array(first_array, second_array, third_array)
	{23, 25, 27, 29, 31, 33, 35, 37, 39, 41 }
	{49, 47, 45, 43, 41, 39, 37, 35, 33 , 31}
	third_array = {33, 35, 37, 39, 41}
	third_array = {0, 0, 0, 0, 0, !, #, %, '<')}
3	ext(third_array)
	min = 33; max = 33
	max = 41; min = 33; (Після перевірки масива)
	Виведення max - min (= 8)
4	out_arr(first_array, second_array, third_array)
	Виведення first_array, second_array, third_array
	Кінець

Висновок.

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

Склали програму раціонального пошуку різниці між максимальним та мінімальним елементами, які створені в масиві, який у свою чергу згенерований на основі інших символьних масивів. Також ми навчилися за допомогою коду створювати символьні масиви, які мають свої властивості та набули навичок використовувати ці властивості.

Лабораторна робота мала особливості щодо формування масиву, де символьні елементи виходять за проміжок символьного елемента. Завдяки лабораторній ми змогли навчитися раціонально розв'язувати такі завдання.