Звіт

# Виконання лабораторної роботи №3

Студента 2ПІ-25Б Заболотного Олександра Івановича

«Робота з одномірними масивами»

Тема: Робота з одновимірними масивами

Мета: Ознайомитись із простими операціями над масивами: ініціалізація масивів, виконання обчислень з масивами, сортування та пошук.

Завдання 8. Сформувати масив IМ(100), елементами якого є числа 2, 1, 4,   
3, 6, 5, .... 100, 99 .

Текст програми:

#include <stdio.h>

int main() {

    int IM[100];

    for (int i = 0; i < 100; i++) {

        if(i%2==0){

            IM[i]=i+2;

        } else {

            IM[i]=i;

        }

    }

    for (int i = 0; i < 100; i++) {

        printf("%d ", IM[i]);

    }

    return 0;

}

1. Призначення програми:

Програма призначена для заповнення масиву цілих чисел за певним правилом: якщо індекс парний — елементу присвоюється значення індексу плюс 2, якщо непарний — значення індексу. Далі всі елементи масиву виводяться на екран.

2. Вхідні дані:

Вхідні дані не вводяться користувачем — програма працює з масивом розміром 100 елементів.

3. Алгоритм роботи програми:

• Оголошення масиву IM[100].

• Заповнення масиву:

 – якщо індекс парний, IM[i] = i + 2

 – якщо індекс непарний, IM[i] = i

• Виведення всіх елементів масиву на екран.

4. Опис основних частин коду:

• #include <stdio.h> — підключення бібліотеки для введення/виведення.

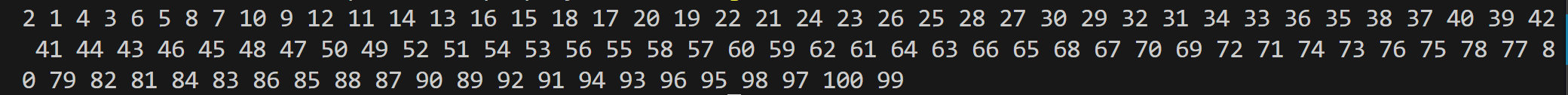
• Оголошення масиву IM[100].

• Цикл for для заповнення масиву згідно з умовою.

• Цикл for для виведення елементів масиву.

• Виведення результату на екран.

Результат роботи:



Висновок: У ході виконання лабораторної роботи я навчився оголошувати масиви, використовувати цикли та умовні оператори мовою C для заповнення і обробки даних.

В результаті я створив програму, яка заповнює масив за заданим правилом і виводить його елементи на екран.

Отримані знання та навички стануть основою для подальшого вивчення програмування та розробки більш складних програм.

Контрольні питання:  
8. Типи масивів у мові C:

За розмірністю:

* Одновимірні масиви: int arr[10];
* Багатовимірні масиви: int matrix[5][3];
* За способом розміщення в пам'яті:
* Статичні масиви (розмір відомий на етапі компіляції): int arr[100];
* Динамічні масиви (розмір визначається під час виконання): через malloc()

За типом елементів:

* Масиви цілих чисел: int arr[10];
* Масиви дійсних чисел: float arr[10];
* Масиви символів (рядки): char str[50];
* Масиви покажчиків: int \*ptrs[10];

9. Оголошення одновимірного масиву через покажчики:

Статичне виділення пам'яті:

int \*ptr;

int arr[10];

ptr = arr;  // або ptr = &arr[0];

Динамічне виділення пам'яті:

int \*ptr;

int size = 10;

ptr = (int\*)malloc(size \* sizeof(int));

// Не забути звільнити пам'ять: free(ptr);

Доступ до елементів: ptr[i] або \*(ptr + i)

10. Визначення необхідного розміру пам'яті для масиву:

Формула: розмір = кількість\_елементів × sizeof(тип\_елемента)

Щоб отримати розмір масиву, можна скористатися оператором sizeof  
Іншим поширеним способом створення масивів є визначення розміру масиву та додавання елементів пізніше.

Використовуючи цей метод, **ви повинні заздалегідь знати кількість елементів масиву**, щоб програма могла зберігати достатньо пам’яті.