9.Arrays in C

- **THEORY EXERCISE**: Explain the concept of arrays in C. Differentiate between one-dimensional and multi-dimensional arrays with examples.

One-Dimensional Arrays

A **one-dimensional array** (or simply an array) is a linear collection of elements. You can think of it as a list of items, where each item can be accessed using a single index.

Declaration and Initialization

Syntax:

data_type array_name[array_size];

Example:

CopyEdit

```
#include <stdio.h>


int main() {
   int numbers[5] = {10, 20, 30, 40, 50};
   for(int i = 0; i < 5; i++) {
      printf("%d\n", numbers[i]);
   }
}
```

Multi-Dimensional Arrays

A **multi-dimensional array** is an array of arrays. The most common type is the two-dimensional array, which can be thought of as a table with rows and columns.

Declaration and Initialization

Syntax:

data_type array_name[size1][size2];

Key Differences Between One-Dimensional and Multi-Dimensional Arrays

Example (2D Array):

CopyEdit

```
#include <stdio.h>
```

```
int main() {
    int matrix[2][3] = {
        {1, 2, 3},
        {4, 5, 6}
    };

    for(int i = 0; i < 2; i++) {
        for(int j = 0; j < 3; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}
```

Feature	**One-Dimensional Array**	**Multi-Dimensional Array**
Structure	Linear (single row)	Tabular (multiple rows and columns)
Declaration Syntax	data_type array_name[size];	data_type array_name[size1][size2];
Accessing Elements	array_name[index];	array_name[index1][index2];
Example	int arr[5];	int matrix[3][4];
Use Case	Storing a list of items	Storing data in a grid or table format