Arrays in C Programming

Introduction

In C programming, an array is a collection of elements of the same data type stored in contiguous memory locations. Arrays provide a convenient way to store and manipulate groups of related data items.

Declaration and Initialization

Declaration

To declare an array in C, you specify the data type of the elements and the size of the array. The general syntax for declaring an array is as follows:

cCopy code

data_type array_name[array_size];

Here, **data_type** specifies the type of elements in the array, **array_name** is the name of the array, and **array_size** specifies the number of elements in the array.

Initialization

You can initialize an array at the time of declaration or later in the program. When initializing an array, you can provide a list of values enclosed in curly braces **{}**.

cCopy code

int numbers[5] = {1, 2, 3, 4, 5};

Accessing Array Elements

You can access individual elements of an array using the array index. Array indices in C start from 0. For example, to access the third element of an array **numbers**, you would use:

cCopy code

int third_element = numbers[2]; // Index 2 corresponds to the third element

Array Operations

Traversing an Array

Traversing an array involves accessing each element of the array one by one. This is commonly done using loops such as **for** or **while**.

cCopy code

int i; for (i = 0; i < 5; i++) { printf("%d ", numbers[i]); }

Modifying Array Elements

You can modify the elements of an array by assigning new values to them.

cCopy code

numbers[2] = 10; // Assigning a new value to the third element

Multidimensional Arrays

C also supports multidimensional arrays, which are arrays of arrays. They are useful for representing matrices and tables.

cCopy code

int matrix[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

Conclusion

Arrays are a fundamental concept in C programming, providing a way to work with collections of data efficiently. Understanding how to declare, initialize, and manipulate arrays is essential for writing C programs effectively.