Array in C

An array is a collection of data items of the same type stored in consecutive memory locations. In the C programming language, arrays are a derived data type that can hold primitive data types such as int, char, double, float, and more. Additionally, arrays can store collections of derived data types like pointers and structures. It is the most basic data structure, allowing random access to each element by referencing its index number.

Syntax for Declaring an Array:

data_type array_name[array_size];

- **data_type:** The type of elements to be stored (e.g., int, float, char).
- **array_name:** The name of the array.
- **array_size:** The number of elements in the array.

Array Initialization in C

Initialization in C refers to the process of assigning an initial value to a variable. When an array is declared or memory is allocated for it, the array's elements hold undefined, garbage values by default. Therefore, it is necessary to initialize the array with meaningful values. In C, there are several methods to initialize an array.

- **Array Initialization with Declaration**

In this method, we initialize the array along with its declaration. We use an initializer list to initialize multiple elements of the array. An initializer list is the list of values enclosed within braces { } separated by comma.

Syntax: data_type array_name [size] = {value1, value2, ... valueN};

```
int Arr[5]={2,4,6,8,10,12};
```

- **Array Initialization with Declaration without Size**

If we initialize an array using an initializer list, we can skip declaring the size of the array as the compiler can automatically deduce the size of the array in these cases. The size of the array in these cases is equal to the number of elements present in the initializer list as the compiler can automatically deduce the size of the array.

Syntax: data_type array_name[] = {value1, value2,....valueN};

```
int Arr[]={2,4,6,8,10,12};
```

- **Array Initialization after Declaration (Using Loops)**

We initialize the array after the declaration by assigning the initial value to each element individually. We can use for loop, while loop, or do-while loop to assign the value to each element of the array.

Syntax:

```
for (int i = 0; i < N; i++) {
   array_name[i] = value i;
}
```

Properties of an Array

- An array has the following characteristics:
- All elements in an array are of the same data type and occupy equal space in memory (e.g., an int takes up 4 bytes).
- The elements are stored in consecutive memory locations, with the first element placed at the smallest memory address.
- Each element can be accessed directly because the address of any element can be determined using the base address and the size of the data type.

Example of array in c:

```
#include <stdio.h>
int main() {
    // Declare and initialize an array
    int numbers[5] = {10, 20, 30, 40, 50};
  // Accessing and printing elements in the array
    for (int i = 0; i < 5; i++) {
        printf("Element at index %d: %d\n", i, numbers[i]);
    }
  return 0;
}
```

Types of Array in C

There are two types of arrays based on the number of dimensions it has. They are as follows:

- One Dimensional Arrays (1D Array)
- Multidimensional Arrays

One Dimensional Array

A one-dimensional array in C is a collection of elements of the same data type, stored in contiguous memory locations. It is essentially a list of variables of the same type that can be accessed using a single index. The index starts from 0, meaning the first element is at index 0, the second at index 1, and so on.
Syntax: data_type array_name[size];
Example: #include <stdio.h>

```
int main() {
    // Declare and initialize a one-dimensional array
    int numbers[5] = {10, 20, 30, 40, 50};
  // Access and display elements using a loop
    printf("Elements of the array:\n");
```

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

    return 0;
}
```

Multidimensional Array in C

A multidimensional array in C is an array of arrays, where each element of an array can itself be another array. The most common type of multidimensional array is a two-dimensional array, but C allows arrays of any dimension.

Syntax: data_type array_name[size1][size2]...[sizeN];

Example:

```
#include <stdio.h>
int main() {
    // Declare and initialize a 2D array (3x3 matrix)
    int matrix[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };
   // Access and display the elements of the 2D array
    printf("Elements of the 2D array:\n");
    for(int i = 0; i < 3; i++) {
        for(int j = 0; j < 3; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n"); // For new line after each row
    }
  return 0;
}
```