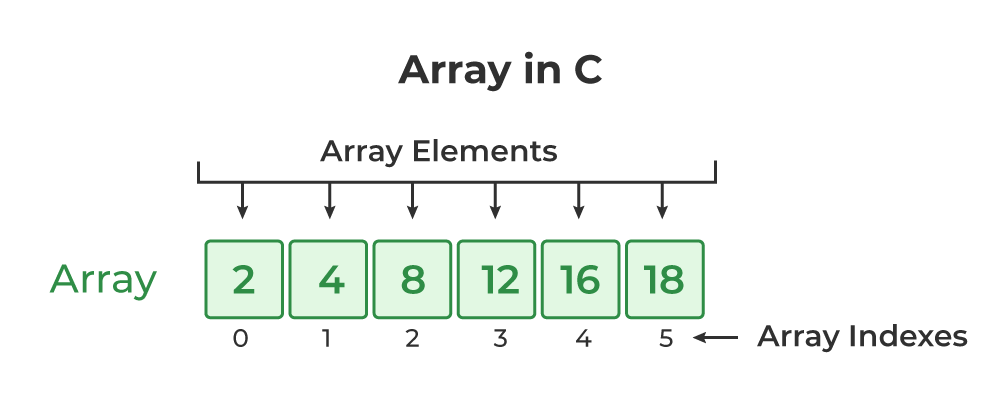
**C Arrays**

**Array in C**is one of the most used data structures in C programming. It is a simple and fast way of storing multiple values under a single name. In this article, we will study the different aspects of array in C language such as array declaration, definition, initialization, types of arrays, array syntax, advantages and disadvantages, and many more.

**What is Array in C?**

An array in C is a fixed-size collection of similar data items stored in contiguous memory locations. It can be used to store the collection of primitive data types such as int, char, float, etc.



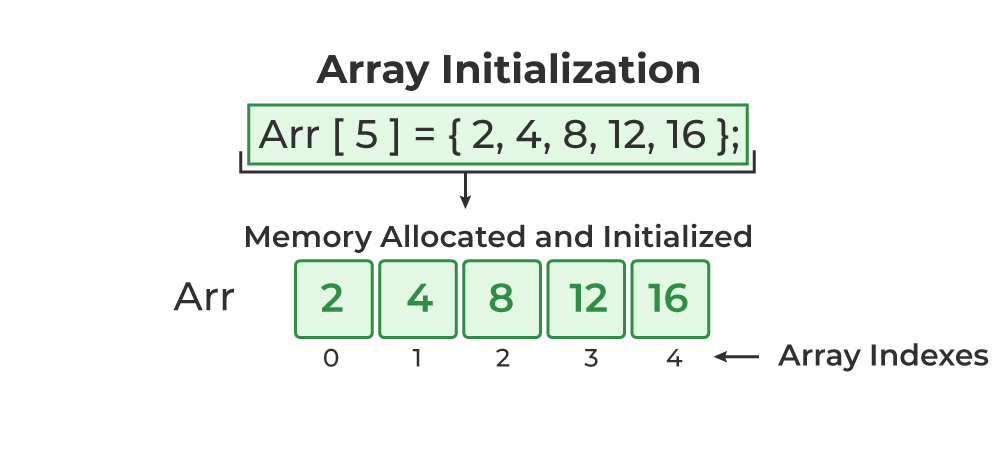
## C Array Initialization

Initialization in C is the process to assign some initial value to the variable. When the array is declared or allocated memory, the elements of the array contain some garbage value. So, we need to initialize the array to some meaningful value. There are multiple ways in which we can initialize an array in C.

### **1. 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 b a comma.

data\_type array\_name [size] = {value1, value2, ... valueN};



### **2. 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.

data\_type array\_name[] = {1,2,3,4,5};

The size of the above arrays is 5 which is automatically deduced by the compiler.

### **3. 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.

for (int i = 0; i < N; i++) {  
 array\_name[i] = valuei;  
}

## Access Array Elements

We can access any element of an array in C using the array subscript operator **[ ]** and the index value***i***of the element.

array\_name [index];

One thing to note is that the indexing in the array always starts with 0, i.e., the **first element** is at index **0** and the **last element** is at **N – 1**where **N** is the number of elements in the array.

**C Program to find the largest number in an array**

#include <stdio.h>

int main()

{

int size, i, largest;

printf("\n Enter the size of the array: ");

scanf("%d", &size);

int array[size]; //Declaring array

//Input array elements

printf("\n Enter %d elements of the array: \n", size);

for (i = 0; i < size; i++)

{

scanf(" %d", &array[i]);

}

//Declaring Largest element as the first element

largest = array[0];

for (i = 1; i < size; i++)

{

if (largest < array[i])

largest = array[i];

}

printf("\n largest element present in the given array is : %d", largest);

return 0;

}

**Sum of all the Array elements**

#include <stdio.h>

int main() {

int n, sum = 0;

// Input the size of the array

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

// Input array elements

printf("Enter %d elements:\n", n);

for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

sum += arr[i];

}

// Display the sum

printf("Sum of elements: %d\n", sum);

return 0;

}

**Array Reverse**

#include <stdio.h>

int main() {

int n;

// Input the size of the array

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

// Input array elements

printf("Enter %d elements:\n", n);

for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

// Reverse the array

for (int i = 0, j = n - 1; i < j; i++, j--) {

// Swap elements at positions i and j

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

// Display the reversed array

printf("Reversed array: ");

for (int i = 0; i < n; i++) {

printf("%d ", arr[i]);

}

return 0;

}

**Array Searching**

#include <stdio.h>

int main() {

int n, key;

// Input the size of the array

printf("Enter the size of the array: ");

scanf("%d", &n);

int arr[n];

// Input array elements

printf("Enter %d elements:\n", n);

for (int i = 0; i < n; i++) {

scanf("%d", &arr[i]);

}

// Input the element to search

printf("Enter the element to search: ");

scanf("%d", &key);

// Search for the element

int index = -1;

for (int i = 0; i < n; i++) {

if (arr[i] == key) {

index = i;

break;

}

}

// Display the result

if (index != -1) {

printf("Element found at index: %d\n", index);

} else {

printf("Element not found in the array\n");

}

return 0;

}