**Array:**

What Are Arrays in Data Structures?

An array is **a linear data structure that collects elements of the same data type and stores them in contiguous and adjacent memory locations**.

Arrays work on an index system starting from 0 to (n-1), where n is the size of the array.

**Array Representation:**

Arrays can be declared in various ways in different languages. For illustration, let's take C array declaration.



As per the above illustration, following are the important points to be considered.

* Index starts with 0.
* Array length is 10 which means it can store 10 elements.
* Each element can be accessed via its index. For example, we can fetch an element at index 6 as 9.

1. **TRAVERSING:**

This operation is used to print all the array elements one by one.

OR

we can say that simply visiting each element one by one is traversing.

**Looping through the elements of an array** is called a traversal. Another common pattern is a search, which involves traversing an array looking for a particular element.

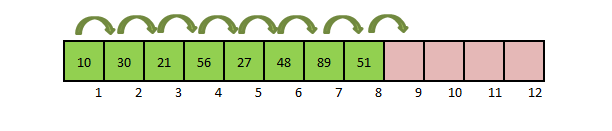
1. **INSERTION:**

Insert operation is to insert one or more data elements into an array. Based on the

requirement, a new element can be added at the beginning, end, or any given index of array.

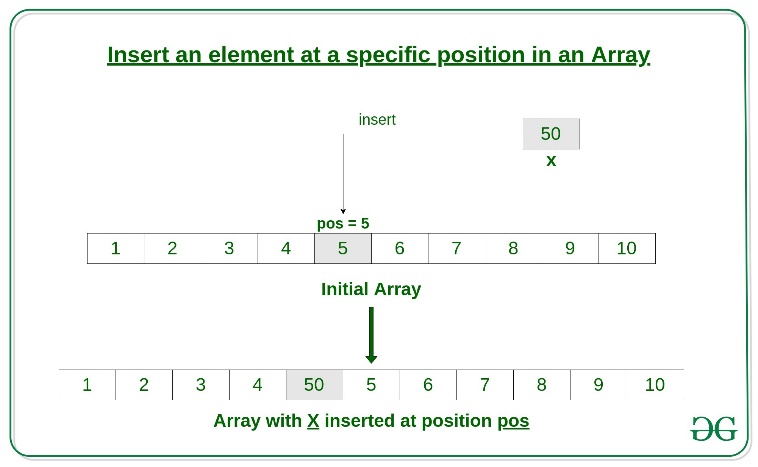
* **INSERTION\_AT\_START:**

In this case we have to move all the elements one position backwards to make  a hole at the beginning of array. Though the insertion process is not difficult but freeing the first location for new element involves movement of all the existing elements. This is the worst case scenario in insertion in a linear array.



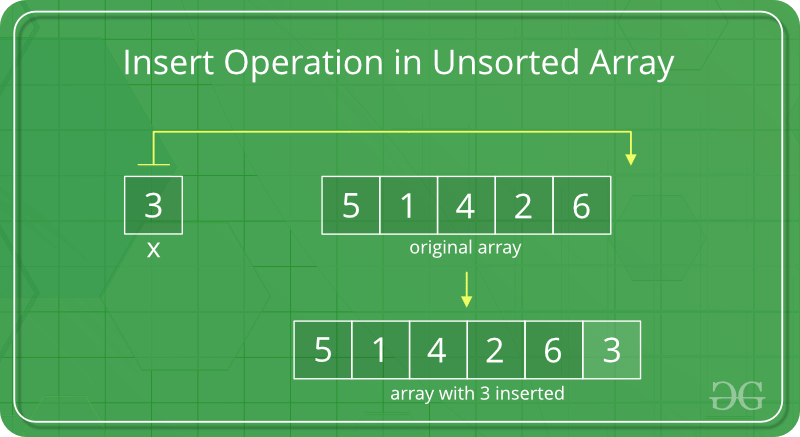
* **INSERTION\_AT\_MIDDLE:**

The following figure shows how a value in inserted in a specific location.



* **INSERTION\_AT\_END:**

In an unsorted array, the insert operation is faster as compared to a sorted array because we don’t have to care about the position at which the element is to be placed. The following figure shows how a value in inserted in a specific location.



1. **UPDATION:**

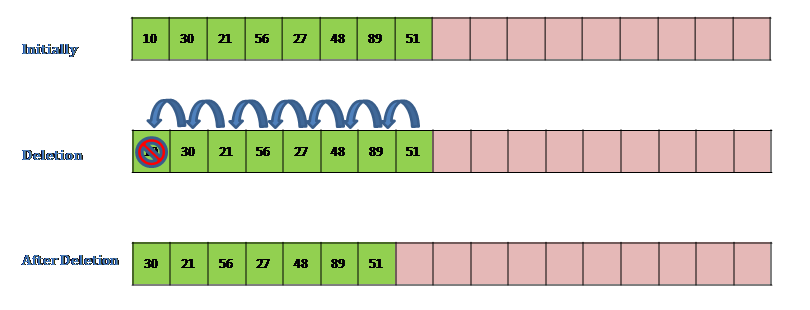
Updating an element in an array means changing the value of an existing element in the array to a new value. This can be done by finding the index of the element to be updated and then updating the value at that index to the new value.

1. **DELETION:**

In the delete operation, the element to be deleted is searched using the [linear search](https://www.geeksforgeeks.org/linear-search/), and then the delete operation is performed followed by shifting the elements

* **DELETION\_AT\_START:**

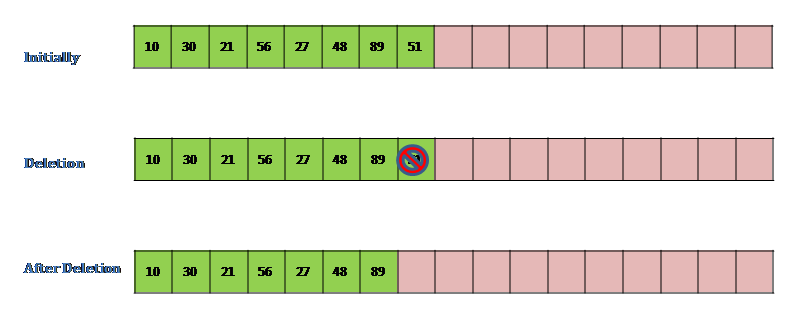
Though the deletion process is not difficult but **moving all elements one position forward involve movement of all the existing elements except the one being deleted**. This is the worst case scenario in deletion in a linear array.



Deletion in Array – an element at the end of the array

In this case we don’t have to move any elements since the action here will be just removing the last element. This is done by redefining the index of last element of linear array = N-1.  This is the best case scenario in deletion in a linear array.

In the example array no elements are moved. The last element is removed by setting the index of last element as 7



Deletion in Array – an element at the give position J

Let J be any location in the array for one existing element. We have to delete the element at J position. To do this starting from J every element is moved one place forward so that the element after index J comes to position of Jthelement. This is the average case scenario in deletion in  linear array.

In the example array ,elements from index J (4) to index 8 have to moved one position forward so that an element at index 4 is replaced by element at index 5. Similarly element at 6th postion comes at 5th position, element at 7th position comes at 6th position and element at 8th position replaces element at 7th position completing the deletion process

