

## Array Insertion and Deletion

- Insert an element into an array at a specified location
- Delete an element from an array at a specified location
- Delete an element having a specified value from an array

### 1. Add an element into an array at a specified location

With the following array named “myArray”, the current number of items in the array is 6. Now, add an element 20 at location 2 of the array by calling the function

**Insert(myArray, 6, 20, 2);**

(remember the index of an array starts at 0)

0	1	2	3	4	5	6	7	8	9
5	14	3	50	7	8				

After the function call, the array looks like this:

5	14	20	3	50	7	8			
---	----	----	---	----	---	---	--	--	--

The number of elements in the array is increased to 7.

```
// insert an item at a specified location in the array
// list (IN/OUT) : the list of movies is updated by adding a new movie at location "location"
// number (IN/OUT): the number is increased by 1 after a successful insertion. If the location of
insertion is not valid, no value is inserted, number remains the same as before
// toAdd (IN): the new movie title to be added in the list
// location (IN): The location in the list where the new movie title is to be inserted
```

**void Insert(int list[], int &number, int toAdd, int location)**

```
{
    int i;

    // check the location is in the valid index range
    if ((location >= 0 && location <= number)) {
        // shift all the values to accomodate the new item
        for (i = number; i > location; i--) {
            list[i] = list[i-1];
        }
        // add the new item
        list[i] = toAdd;
        number = number + 1;
    }
    else
    {
        cerr << "The location is out of boundary" << endl;
        cerr << "new item can not be added" << endl;
    }
}
```

## 2. Delete an element from an array at a specified location

With the following array named “myArray”, the current number of items in the array is 6. Now, remove an element at location 1 of the array by calling the function

**Delete(myArray, 6, 1);**

0	1	2	3	4	5	6	7	8	9
5	14	3	50	7	8				

After the function call, the array looks like this:

5	14	50	7	8	8				
---	----	----	---	---	---	--	--	--	--

The number of elements in the array is reduced to 5.

```
// delete an item at a specified location in the array
// list (IN/OUT) : The list of items
// number (IN/OUT) : the number of items in the list,
//                  will increase by 1 if a successful insertion is performed
// location (IN) : The location of the item to be deleted in the list
```

```
void Delete(int list[], int &number, int location)
{
    int i;

    // check the location is in the valid index range
    if ((location >= 0 && location < number)) {
        // remove the item by shifting the items
        for (i = location; i < number - 1; i++) {
            list[i] = list[i + 1];
        }
        number = number - 1;
    }
    else {
        cerr << "The location is out of boundary." << endl;
        cerr << "The item can not be deleted." << endl;
    }
}
```

## 3. Delete an element having a specified value from an array

0	1	2	3	4	5	6	7	8	9
5	14	3	50	7	8				

With array “myArray” as specified before, what if one wants to remove the item having value “50” from the array? or having value “60” from the array?

Write the C++ function to accomplish this.