Remove duplicates from a sorted array

Given a sorted array, the task is to remove the duplicate elements from the array.

Examples:

```
Input  : arr[] = {2, 2, 2, 2, 2}
Output : arr[] = {2}
         new size = 1

Input  : arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5}
Output : arr[] = {1, 2, 3, 4, 5}
         new size = 5
```

Method 1: (Using extra space)

1. Create an auxiliary array temp[] to store unique elements.
2. Traverse input array and one by one copy unique elements of arr[] to temp[]. Also keep track of count of unique elements. Let this count be **j**.
3. Copy **j** elements from temp[] to arr[] and return j

```
// Simple C++ program to remove duplicate
#include <iostream>
using namespace std;

// Function to remove duplicate elements This function
// returns new size of modified array.
int removeDuplicates(int arr[], int n)
{
    // Return, if array is empty or contains a single
    // element    \
   if (n == 0 || n == 1)
        return n;

    int temp[n];
```

```
    // Start traversing elements
    int j = 0;
    // If current element is not equal to next element
    // then store that current element
    for (int i = 0; i < n - 1; i++)
        if (arr[i] != arr[i + 1])
            temp[j++] = arr[i];

    // Store the last element as whether it is unique or
    // repeated, it hasn't stored previously
    temp[j++] = arr[n - 1];

    // Modify original array
    for (int i = 0; i < j; i++)
        arr[i] = temp[i];

    return j;
}

// Driver code
int main()
{
    int arr[] = { 1, 2, 2, 3, 4, 4, 4, 5, 5 };
    int n = sizeof(arr) / sizeof(arr[0]);

    // removeDuplicates() returns new size of array.
    n = removeDuplicates(arr, n);

    // Print updated array    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

Output:

```
1 2 3 4 5
```

Time Complexity : O(n)

Auxiliary Space : O(n)

Method 2: (Constant extra space)

Just maintain a separate index for same array as maintained for different array in Method 1.

```
// C++ program to remove duplicates in-place
#include<iostream>
using namespace std;

// Function to remove duplicate elements
// This function returns new size of modified
// array.int removeDuplicates(int arr[], int n)
{
    if (n==0 || n==1)
        return n;

    // To store index of next unique element    int j = 0;

    // Doing same as done in Method 1
    // Just maintaining another updated index i.e. j
    for (int i=0; i < n-1; i++)
        if (arr[i] != arr[i+1])
            arr[j++] = arr[i];

    arr[j++] = arr[n-1];

    return j;
}

// Driver codeint main()
{
    int arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5};
    int n = sizeof(arr) / sizeof(arr[0]);

    // removeDuplicates() returns new size of
    // array.
    n = removeDuplicates(arr, n);
```

```
    // Print updated array
    for (int i=0; i<n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

Output:

```
1 2 3 4 5
```

Time Complexity : O(n)

Auxiliary Space : O(1)