Linear Search

Program A

```
#include <iostream>
using namespace std;

int main()
{
 int arr[10] = {69, 39, 29, 44, 10, 56, 40, 24, 13, 51};
 int val = 56;
 int index = -1;

 for (int i = 0; i < 10; i++)
  {
     if (arr[i] == val)
                 {
                         index = i;
                         break;
                 }
  }

 cout<<"The elements of the array are - ";
 for (int i = 0; i < 10; i++)
  cout<<arr[i]<<" ";

 cout<<"\nThe Element to be searched is - "<<val;

 if (index == -1)
  cout<<"\nElement is not present in the array";
 else
  cout<<"\nElement is present at "<<index<<"
position of array";

 return 0;
}
```

Program B

```
#include<iostream>
using namespace std;

int main()
{
   int arr[100], i, n,num, index = -1;
   Cout<<"Enter the size of array: max is 100";
   cin>>n;

   cout<<"Enter "<<n<< "Numbers: ";
   for(i=0; i<n; i++)
      cin>>arr[i];

   cout<<"\nEnter a Number to Search: ";
   cin>>num;

   for(i=0; i<n; i++)
   {
      if(arr[i]==num)
      {
         index = i;
         break;
      }
   }

 if (index==-1)
   cout<<"\nThe item is not found in the Array!"<<endl;
else
  cout<<"\nThe item is found at Index No."<<index<<endl;

   return 0;
   }
```

Binary Search

Program C

```
#include<iostream>
using namespace std;
int main()
{
   int i, j, temp, num = 9;
   int arr[] = {5, 12, 3, 9, 8, 2, 1};
   int n = sizeof(arr) / sizeof(arr[0]);
   int first = 0, last = (n-1);
   int middle = (first+last)/2;

// print array elements
   cout<<"\nThe Array elements before sorted:\n";
   for(i=0; i<n; i++)
      cout<<arr[i]<<" ";

// sort the array first
   for(i=0; i<n; i++) {
      for(j=0; j<(n-i-1); j++) {
         if(arr[j]>arr[j+1])
         {
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
         }
      }
   }
// print the sorted array
   cout<<"\nThe New Sorted Array is:\n";
   for(i=0; i<n; i++)
      cout<<arr[i]<<" ";

   cout<<"The number to be searched is: "<<num<<endl;

  while(first <= last) {
      if(arr[middle]<num)
         first = middle+1;
      else if(arr[middle]==num)
      {
         cout<<"\nThe number, "<<num<<" found at Position "<<middle;
         break;
      }
      else
         last = middle-1;

      middle = (first+last)/2;
   }
   if(first>last)
      cout<<"\nThe number, "<<num<<" is not found in the given Array"<<endl;
   return 0;
   }
```

Program D

```
#include<iostream>
using namespace std;

//Function declaration
void sortArray(int [], int);
int binarySearch(int [], int, int);

int main()
{
   int n, i, arr[50], num, result;

   cout<<"Enter the Size (max. 50): ";
   cin>>n;

//Accepting array elements from the user
   cout<<"Enter "<<n<<" Elements: ";
   for(i=0; i<n; i++)
      cin>>arr[i];

 // sort the array first
   sortArray(arr, n);

 // Displaying the sorted array list
   cout<<"\nThe New Sorted Array is:\n";
   for(i=0; i<n; i++)
      cout<<arr[i]<<" ";

   cout<<"\n\nEnter the Element to be Searched: ";
   cin>>num;

  // search the element using binary search
   result = binarySearch(arr, num, n);

  if(result==0)
      cout<<endl<<num<<" is not available in the list";
   else
     cout<<endl<<num<<" is available at Position "<< result;

   cout<<endl;

   return 0;
}
```

```
//sortArray function definition
void sortArray(int arr[], int n)
{
   int i, j, temp;
   for(i=0; i<n; i++)
   {
      for(j=0; j<(n-i-1); j++)
      {
         if(arr[j]>arr[j+1])
         {
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
         }
      }
   }
}

//binarySearch function definition
int binarySearch(int arr[], int num, int n)
{
   int first, last, middle;
   first = 0;
   last = (n-1);
   middle = (first+last)/2;
   while(first <= last)
   {
      if(arr[middle]<num)
         first = middle+1;
      else if(arr[middle]==num)
         return middle;
      else
         last = middle-1;

      middle = (first+last)/2;
   }
   return 0;
}
```

Exercise:

Write a C++ program to search a duplicated elements from the given array or list.