```
public class Answer {

  //Method to sort the array
  public static void bubbleSort(int arr[]){

    int i,j,temp;
    boolean swapped;

    for(i=0; i < arr.length - 1; i++){
      swapped = false;
      for(j=0; j < arr.length - i - 1; j++){
        if(arr[j] > arr[j+1]){
          temp = arr[j];
          arr[j] = arr[j+1];
          arr[j+1] = temp;
          swapped = true;
        }
      }

      if(swapped == false){
        break;
      }
    }
  }

  //Method to search
  static int binarySearch(int arr[], int x) {
    int low = 0, high = arr.length - 1;
    while (low <= high) {
```

```
        int mid = low + (high - low) / 2;

        // Check if x is present at mid
        if (arr[mid] == x)
            return mid;
        // If x is greater, ignore the left half
        if (arr[mid] < x)
            low = mid + 1;
        // If x is smaller, ignore the right half
        else
            high = mid - 1;
    }
    // If we reach here, then the element was not present
    return -1;
}

// Prints the array
static void printArray(int arr[]) {
    int n = arr.length;
    for (int i=0; i<n; ++i)
        System.out.print(arr[i]+" ");
    System.out.println();
}

public static void main(String[] args) {
        int arr[] = {641,1,57,5,200,45,132,223,8};
        int x = 31;

        bubbleSort(arr);
```

```
        printArray(arr);

        int result = binarySearch(arr,x);

        if(result == -1){
           System.out.println("Element is not present in the array");
        }
        else{
           System.out.println("Element is present at index: "+result);
        }
    }
}
```

Linear Search Algorithm

```
import java.io.*;

class GFG {
   public static int search(int arr[], int N, int x)
   {
      for (int i = 0; i < N; i++) {
         if (arr[i] == x)
            return i;
      }
      return -1;
   }
```