EX.No:1 (i)

```
class BinarySearchExample{
 public static void binarySearch(int arr[], int first, int last, int key){
   int mid = (first + last)/2;
   while( first <= last ){
      if ( arr[mid] < key ){
        first = mid + 1;
      }else if ( arr[mid] == key ){
        System.out.println("Element is found at index: " + mid);
        break;
      }else{
         last = mid - 1;
      }
      mid = (first + last)/2;
   }
   if ( first > last ){
      System.out.println("Element is not found!");
   }
 }
 public static void main(String args[]){
        int arr[] = {10,20,30,40,50};
        int key = 30;
        int last=arr.length-1;
        binarySearch(arr,0,last,key);
 }
}
```

OUTPUT

Compile by: javac BinarySearchExample.java

Run by: java BinarySearchExample

```
Element is found at index: 2
```

EX NO : 1 (ii)

```
import java.util.Scanner;
public class LinearSearchExample
{
public static intlinearSearch(int[] arr, int key){
for(int i=0;i<arr.length;i++){
if(arr[i] == key){
return i;
            }
        }
return -1;
    }
public static void main(String a[]){
int[] a1= {10,20,30,50,70,90};
int key = 50;
System.out.println(key+" is found at index: "+linearSearch(a1, key));
    }
}
```

OUTPUT:

Compile by: javac LinearSearchExample.java

Run by: java LinearSearchExample

```
50 is found at index: 3
```

EX NO :1 (iii)

```
public class InsertionSortExample {
    public static void insertionSort(int array[]) {
        int n = array.length;
        for (int j = 1; j < n; j++) {
            int key = array[j];
            int i = j-1;
            while ( (i > -1) && ( array [i] > key ) ) {
                array [i+1] = array [i];
                i--;
            }
            array[i+1] = key;
        }     }
    public static void main(String a[]){
        int[] arr1 = {9,14,3,2,43,11,58,22};
        System.out.println("Before Insertion Sort");
        for(int i:arr1){
            System.out.print(i+" ");
        }
        System.out.println();

        insertionSort(arr1);//sorting array using insertion sort

        System.out.println("After Insertion Sort");
        for(int i:arr1){
            System.out.print(i+" ");
        }       }    }
```

OUTPUT:

```
Before Insertion Sort
9 14 3 2 43 11 58 22
After Insertion Sort
2 3 9 11 14 22 43 58
```

EX NO: 1(iv)

```
public class SelectionSortExample {
    public static void selectionSort(int[] arr){
        for (int i = 0; i < arr.length - 1; i++)
        {
            int index = i;
            for (int j = i + 1; j < arr.length; j++){
                if (arr[j] < arr[index]){
                    index = j;//searching for lowest index
                }             }
            int smallerNumber = arr[index];
            arr[index] = arr[i];
            arr[i] = smallerNumber;
        }      }
    public static void main(String a[]){
        int[] arr1 = {9,14,3,2,43,11,58,22};
        System.out.println("Before Selection Sort");
        for(int i:arr1){
            System.out.print(i+" ");
        }
System.out.println();
  selectionSort(arr1);//sorting array using selection sort
  System.out.println("After Selection Sort");
  for(int i:arr1){
    System.out.print(i+" ");
        }      } }
```

OUTPUT:

```
Before Selection Sort

9 14 3 2 43 11 58 22

After Selection Sort
2 3 9 11 14 22 43 58
```