Q1.SHOW the elements of the following array after 4 passes of bubble sort.

40 20 50 60 30 10

Q2.SHOW the elements of the following array after 4 passes of selection sort.

82 42 49 8 25 52 36 93 59

Q3.
Q2.SHOW the elements of the following array after 4 passes of insertion sort

82 42 49 8 25 52 36 93 5

Q4.Find the index of first 1 in a sorted array of 0's and 1's
Given a sorted array consisting 0's and 1's. The problem is to find the index of first '1' in the sorted array. It could be possible that the array consists of only 0's or only 1's. If 1's are not present in the array then print "-1".

```
Input : arr[] = {0, 0, 0, 0, 0, 0, 1, 1, 1, 1}
Output : 6
The index of first 1 in the array is 6.

Input : arr[] = {0, 0, 0, 0}
Output : -1
1's are not present in the array.
```

Q5.

Choose the appropriate code that does binary search.

a.

```
public static int recursive(int arr[], int low, int high, int key)

{
        int mid = low + (high - low)/2;
        if(arr[mid] == key)
        {
                return mid;
        }
        else if(arr[mid] < key)
        {
                return recursive(arr,mid+1,high,key);
        }
        else
        {
                return recursive(arr,low,mid-1,key);
        }
}
```

b.

```
public static int iterative(int arr[], int key)

{
        int low = 0;
        int mid = 0;
        int high = arr.length-1;
        while(low <= high)
        {
                mid = low + (high + low)/2;
                if(arr[mid] == key)
                {
                        return mid;
                }
                else if(arr[mid] < key)
                {
                        low = mid + 1;
                }
                else
                {
                        high = mid - 1;
                }
        }
        return -1;


}
```

c.public static int recursive(int arr[], int low, int high, int key)

```
{
        int mid = low + (high + low)/2;
        if(arr[mid] == key)
        {
                return mid;
        }
        else if(arr[mid] < key)
        {
                return recursive(arr,mid-1,high,key);
        }
        else
        {
                return recursive(arr,low,mid+1,key);
        }
}
```

Q6. An array containing 7 elements
12 19 23 27 34 45

perform binary search to find 27.