```
//Find the peak element
let a = [1,2,3,1];


const func = (array)=> {

   let low = 0;
   let high = array.length-1;
   let mid;

   while(low <= high){

       mid = Math.floor(low+(high-low)/2);

       if(low == high)
        return low;

        else if(array[mid] < array[mid+1])
        low = mid+1;

        else {
            high = mid;
        }
   }
}
//TC : O(logn)
//SC: O(1)
console.log(func(a));
```

```
//2. Two Sum II - Input Array Is Sorted
let arr = [2,4,5,6,7,8,16];
let target = 14;
//return an array with index -> [3,5]

//Brute Force O(n^2)
```

```
//Two Pointer O(n)

//Binary Search
//TC: O(n* logn)

const func = (arr, target) => {

    for(let i=0;i<arr.length;i++){

        let complement = target - arr[i];

       let low = i+1;
       let high = arr.length-1;

       let mid;
       while(low<=high){

           mid = Math.floor(low+(high-low)/2);

           if(arr[mid] == complement)
           {
              return [i,mid];
           }

           else if(arr[mid]>complement){
               high = mid-1;
           }

           else{
               low = mid+1;
           }
       }
    }
}

console.log(func(arr, target));
```

```
// Search in Rotated Sorted Array II

let arr = [1,0,1,1,1];
let target = 0

//Linear Search -> for loop -> O(n)

//Binary Search
const func = (arr, target) => {

    let low =0;
    let high = arr.length -1;

    let mid;

    while(low<= high){

        mid = Math.floor(low+(high-low)/2);

        if(arr[mid] === target)
            return true;

        else if(arr[mid] < arr[high] || arr[low] > arr[mid])
        {
            if(target > arr[mid] && target <=arr[high])
                low = mid+1;
            else
            {
                high = mid-1;
            }
        }

        else if(arr[low] < arr[mid] || arr[mid] > arr[high])
        {
            if(target >= arr[low] && target < arr[mid])
                high = mid-1;
            else
             low = mid+1;
        }

       else
       {
           // arr[low] === arr[high] === arr[mid]
           low++;
```

```
      }
    }
  return false;
}


console.log(func(arr, target));
```