Searching in an array- Linear & Binary Search

Q1. Given an array. Find the number X in the array. If the element is present, return the index of the element, else print “Element not found in array”. Input the size of array, array from user and the element X from user. Use Linear Search to find the element.

```java
import java.util.Scanner;

public class q1 {
    static int sreach(int arr[], int x){

        for(int i = 0; i < arr.length; i++){
            if(arr[i] == x){
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Array size");
        int s = sc.nextInt();

        int arr[] = new int[s];

        System.out.println("enter elements of array");
        for(int i = 0; i < arr.length; i++){
            arr[i] = sc.nextInt();
        }
        System.out.println("Enter element to sreach");
        int x = sc.nextInt();

        int result = sreach(arr, x);

        if(result == -1){
            System.out.println("Element not found");
        }
        else{
            System.out.printf("Found %d at index %d\n", x,
result );
        }
    }
}
```

Q2. Given an array and an integer "target", return the last occurrence of "target" in the array. If the target is not present return -1.

```
import java.util.Scanner;

public class q2 {

    private static int lastOccurrence(int[] arr, int x) {

        int start = 0, end = arr.length - 1;
        int lo = -1;
        while(start <= end){
            int mid = start + (end - start)/2;
            if(arr[mid] == x){
                lo = mid;
                start = mid + 1;
            }
            else if(arr[mid] < x){
                start = mid + 1;
            }
            else{
                end = mid - 1;
            }
        }
        return lo;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Array size");
        int s = sc.nextInt();

        int arr[] = new int[s];

        System.out.println("enter elements of array");
        for(int i = 0; i < arr.length; i++){
            arr[i] = sc.nextInt();
        }
        System.out.println("Enter element to sreach last
occurrence");
        int x = sc.nextInt();

        int result = lastOccurrence(arr, x);

        if(result == -1){
            System.out.println("Element not found");
        }
        else{
            System.out.printf("last occurrence of %d found 
at index %d\n",
             x, result );
        }
    }
}
```

Q3. Given a sorted binary array, efficiently count the total number of 1's in it.

```
import java.util.Scanner;

public class q3 {

    static int countOne(int arr[]){

        int start = 0, end = arr.length - 1;
        int fo =-1;
        while(start <= end){
            int mid = start + (end - start)/2;
            if(arr[mid] == 1){
                fo = mid;
                end = mid - 1;
            }
            else{
                start = mid + 1;
            }
        }
        return fo;
    }
    public static void main(String[] args) {
```

```java
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Array size");
        int s = sc.nextInt();

        int arr[] = new int[s];

        System.out.println("enter elements of array");
        for(int i = 0; i < arr.length; i++){
            arr[i] = sc.nextInt();
        }
        int c = countOne(arr);
        System.out.println(arr.length - c);
    }
}
```

Q4. Given a sorted integer array containing duplicates, count occurrences of a given number. If the element is not found in the array, report that as well.
Input: nums[] = [2, 5, 5, 5, 6, 6, 8, 9, 9, 9]
target = 5

```java
import java.util.Scanner;

public class q4 {
    static int fisrtOccurrence(int arr[], int x){
        int start = 0, end = arr.length - 1;
        int fo = -1;
        while(start <= end){
            int mid = start + (end - start)/2;
            if(x == arr[mid]){
                fo = mid;
                end = mid - 1;
            }
            else if(x < arr[mid]){
                end = mid - 1;
            }
            else{
                start = mid + 1;
            }
        }
        return fo;
    }

    static int lastOccurrence(int arr[], int x){
        int start = 0, end = arr.length - 1;
        int lo = -1;
        while(start <= end){
            int mid = start + (end - start)/2;
            if(x == arr[mid]){
                lo = mid;
                start = mid + 1;
            }
            else if(x > arr[mid]){
                start = mid + 1;
            }
            else{
                end = mid - 1;
            }
        }
        return lo;
    }



    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Array size");
        int s = sc.nextInt();

        int arr[] = new int[s];

        System.out.println("enter elements of array");
        for(int i = 0; i < arr.length; i++){
            arr[i] = sc.nextInt();
        }

        System.out.println("Enter target to find its
occurrence");
        int x = sc.nextInt();

        int fisrt = fisrtOccurrence(arr, x);
        int last = lastOccurrence(arr, x);

        if(fisrt == last && fisrt == -1){
            System.out.println("Not found");
        }
        else{
            int ans = last - fisrt + 1;
            System.out.println(ans);
        }
    }
}
```

Q5: Given a positive integer num, return true if num is a perfect square or false otherwise.

A perfect square is an integer that is the square of an integer. In other words, it is the product of some integer with itself.
Input: num = 16
Output: true

```java
import java.util.Scanner;

public class q5 {

    static boolean squareRoot(int num){

        int start = 0, end = num/2, result = -1;
        while(start < end){
            int mid = start + (end - start)/2;
            if(mid * mid == num){
                return true;
            }
            else if(mid * mid < num){
                result = mid;
                start = mid + 1;
            }
            else{
                end = mid - 1;
            }
        }
        return false;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Number");
        int num = sc.nextInt();

        System.out.println(squareRoot(num));
    }
}
```