

# Searching in an array- Linear & Binary Search

## Assignment Questions

### Assignment Questions

1. 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.

Ans :

```
import java.util.*;
public class Assignment_q_one {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter the number of elements present in the array");
        int n = sc.nextInt();

        int ar [] = new int [n];

        System.out.println("Enter the elements of the array");

        for(int i=0 ; i< ar.length ; i++){
            ar[i] = sc.nextInt();
        }

        System.out.println("Enter the element to be searched in array");
        int x = sc.nextInt();

        int f = 0;

        for(int i=0 ; i< ar.length ; i++){
```

```

        if (ar[i] == x){
            System.out.println("The element present at the index " + i);
            f++;
            break;
        }
    }

    if(f==0){
        System.out.println("Element not found in array");
    }

}
}

```

**Output :-**

**Enter the number of elements present in the array**

**5**

**Enter the elements of the array**

**6 2 3 1 7**

**Enter the element to be searched in array**

**2**

**The element present at the index 1**

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

**Input 1: arr = [111234456666], target = 4**

**Output 1: 6**

**Input 2: arr = [2226 6 18 29 30 30 30], target = 15**

**Output 2: -1**

**Ans :-**

```

import java.util.*;
public class Assignment_q_two {

    public static int assignment(int ar[] , int x){

        int low = 0;
        int high = ar.length-1;
        int r = -1;
        while(low <= high){
            int mid = (low + high)/2;

```

```

        if(ar[mid] == x){

            r = mid;
            low = mid + 1;
        }
        else if(ar[mid] < x){
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }

    }

    return r;

}

public static void main(String[]args){
    Scanner sc = new Scanner (System.in);
    System.out.println("Enter the number of element present in the array");
    int n = sc.nextInt();

    int ar [] = new int [n];

    System.out.println("Enter the numbers present in the array");
    for(int i=0 ; i<ar.length ; i++){
        ar[i] = sc.nextInt();
    }

    System.out.println("Enter the number which you have to search in this
array");
    int x = sc.nextInt();

    int result = assignment(ar, x);

    if(result == -1){
        System.out.println(result);
    }
    else{
        System.out.println("The number " + x + " is present at the index " +
result);
    }
}

```

```
}
```

Output :-

Enter the number of element present in the array

12

Enter the numbers present in the array

1 1 1 2 3 4 4 5 6 6 6 6

Enter the number which you have to search in this array

4

The number 4 is present at the index 6

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

Input 1: arr = [0000111111]

Output 1: 6

Input 2: arr = [0000011]

Output 2: 2

Ans :-

```
import java.util.*;

public class Assignment_q_three_a {

    public static int numberOfOnes(int ar[]){

        int low = 0;
        int high = ar.length-1;
        int r = -1;

        while(low <= high){
            int mid = (low + high)/2;
            if(ar[mid] == 1){
                r = mid;
                high = mid - 1;
            }
            else{
                low = mid + 1;
            }
        }

        return r;
    }
}
```

```

    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elements present in an array");
        int n = sc.nextInt();

        int ar[] = new int [n];

        System.out.println("Enter the numbers present in this array");
        for(int i=0 ; i<ar.length ; i++){
            ar[i] = sc.nextInt();
        }

        int result = numberOfOnes(ar);

        if(result == -1){
            System.out.println("The number 1 is not present in this array");
        }
        else{
            int p = n - result;

            System.out.println("The number of 1's present in this array is " + p);
        }
    }
}

```

**Ans :-**

**Enter the number of elements present in an array**

**10**

**Enter the numbers present in this array**

**0 0 0 0 1 1 1 1 1 1**

**The number of 1's present in this array is 6**

4. 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**

**Output: Target 5 occurs 3 times**

**Input: nums[] = [2, 5, 5, 5, 6, 6, 8, 9, 9, 9]**

**target = 6**

Output: Target 6 occurs 2 times

Ans :-

```
import java.util.*;
public class Assignment_q_four {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elements present in an array");
        int n = sc.nextInt();

        int ar[] = new int [n];

        System.out.println("Enter the numbers present in this array");
        for(int i=0; i<ar.length ; i++){
            ar[i] = sc.nextInt();
        }

        System.out.println("Enter the number which you have to select in this
array");
        int x = sc.nextInt();

        int low = 0;
        int high = ar.length-1;
        int r = -1;
        while(low <= high){
            int mid = (low + high)/2;
            if (ar[mid] == x){
                r = mid;
                high = mid - 1;

            }
            else if (ar[mid] < x){
                low = mid + 1;
            }
            else {
                high = mid -1;
            }
        }

        low = 0;
        high = ar.length - 1;
        int k = -1;
        while(low <= high){
            int mid = (low + high)/2;
```

```

        if (ar[mid] == x){
            k = mid;
            low = mid + 1;

        }
        else if (ar[mid] < x){
            low = mid + 1;
        }
        else {
            high = mid -1;
        }
    }

    if(r == -1 || k == -1){
        System.out.println("The number " + x + " is not present in this
array");
    }
    else {
        int result = k - r + 1;
        System.out.println("Target " + x + " occurs " + result + " times");
    }

}
}

```

**Output :-**

**Enter the number of elements present in an array**

**10**

**Enter the numbers present in this array**

**2 5 5 5 6 6 8 9 9 9**

**Enter the number which you have to select in this array**

**5**

**Target 5 occurs 3 times**

5. 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.

**Example 1:**

**Input: num = 16**

**Output: true**

**Explanation: We return true because  $4 * 4 = 16$  and 4 is an integer.**

### Example 2:

Input: num = 14

Output: false

Explanation: We return false because  $3.742 * 3.742 = 14$  and 3.742 is not an integer.

Ans :-

```
import java.util.*;
public class Assignment_q_five {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter the number that you want to check whether it is
perfect square or not");
        int s = sc.nextInt();

        int low = 0;
        int high = s;
        int f = 0;
        while (low <= high) {

            int mid = (low + high) / 2;
            int value = mid * mid;

            if(s == value){
                f++;
                System.out.println("true");
                break;
            }
            else if (s < value){
                high = mid - 1;
            }
            else{
                low = mid + 1;
            }
        }

        if(f == 0){
            System.out.println("false");
        }
    }
}
```

Output :-



**Enter the number that you want to check whether it is perfect square or not**

**16**

**true**