

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.

```
import java.io.*;
import java.util.*;
public class Main{

    public static void main(String args[]){

        int m;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of elements you want to add : ");
        m=sc.nextInt();

        int []arr = new int[m];

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

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

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

        int idx = -1;
        for(int i=0;i<m;i++){
            if(arr[i]==element){
                idx = i;
                break;
            }
        }

        if(idx!=-1){
            System.out.println(idx);
        }
        else{
            System.out.println("Element not found in array");
        }
    }
}
```

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.

Input 1: arr = [1 1 1 2 3 4 4 5 6 6 6 6] , target = 4 Output 1: 6

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

Output 2: -1

```
import java.io.*;
import java.util.*;
public class Main{

    public static int numberOf1s(int[] nums, int low, int high){
        while(low <= high){
            int mid = low + (high - low)/2;

            if(nums[mid] == 0){
                low = mid + 1;
            }
            else {
                high = mid - 1;
            }
        }
        return (nums.length - low);
    }

    public static void main(String args[]){

        int m;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of elements you want to add : ");
        m=sc.nextInt();

        int []arr = new int[m];

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

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

        System.out.println("The number of one's in the given array is/are: " +
        numberOf1s(arr , 0 , m - 1));
    }
}
```

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

Input 1: arr = [^ ^ ^ 1 1 1 1 1] Output 1: 6

Input 2: arr = [^ ^ ^ ^ 1 1]

Output 2: 2

```
import java.io.*;
import java.util.*;
public class Main{

    public static int lastOccurrence(int[] a, int low, int high, int target){
        int answer = -1;
        while(low <= high){
            int mid = low + (high - low)/2;

            if(a[mid] == target){
                answer = mid;
                low = mid + 1; //if you found the target or if target is
greater than the current element, to find last occurrence move to right half
of the array
            }
            else if(a[mid] > target){
                high = mid - 1;
            }
            else low = mid + 1;
        }
        return answer;
    }

    public static int firstOccurrence(int[] a, int low , int high , int
target){

        int answer = -1;

        while(low <= high) {

            int mid = (low + high)/2;

            if(a[mid] == target) {
                answer = mid;
                high = mid - 1; // trying to find the minimum index at which
value x is present
            }
            else if(a[mid] > target) {
                high = mid - 1;
            }

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

```

        return answer;
    }

    public static void main(String args[]){

        int m;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of elements you want to add : ");
        m=sc.nextInt();

        int []arr = new int[m];

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

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

        int target;
        Scanner sc1 = new Scanner(System.in);
        System.out.print("enter the target: ");
        target = sc1.nextInt();

        int first = firstOccurrence(arr , 0 , m - 1 , target);
        int last = lastOccurrence(arr , 0 , m - 1 , target);

        if(first == last && first == -1)System.out.println("The target does
not exist in the array.");
        else System.out.println("The frequency of target in the given array is
" + (last - first + 1) + " time/times");
    }
}

```

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

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

```

import java.io.*;
import java.util.*;
public class Main{

```

```

public static int repeatedNumber(int[] a, int low, int high){
    int answer = -1;
    while(low <= high){
        int mid = low + (high - low)/2;

        if(a[mid] > mid)low = mid + 1;
        else {
            answer = a[mid];
            high = mid - 1;
        }
    }
    return answer;
}

public static void main(String args[]){

    int m;
    Scanner sc=new Scanner(System.in);
    System.out.print("Enter the number of elements you want to add : ");
    m=sc.nextInt();

    int []arr = new int[m];

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

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

    System.out.println("The repeated number in the given array is " +
repeatedNumber(arr , 0 , m - 1));
}
}

```

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.

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.

```
import java.io.*;
import java.util.*;
public class Main{

    public static boolean isPerfectSquare(int num) {
        if(num == 1) return true ; // edge case
        long start = 0 ; // Take Long as the inputs have large value
        long end = num /2 ;
        while(start <= end ){
            long mid = start + ( end - start ) / 2 ;
            if (mid * mid > num) end = mid - 1 ; // Mid is greater than the
            squareroot of the number
            else if (mid * mid < num) start = mid + 1 ; // Mid is less than
            our required number
            else return true; // we found our squareroot number
        }
        return false ;
    }

    public static void main(String args[]){
        int m;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number : ");
        m=sc.nextInt();

        System.out.println("The given number is a perfect square: " +
        isPerfectSquare(m));
    }
}
```