

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

Input 1: arr = [0 0 0 0 1 1 1 1 1 1]

Output 1: 6

Input 2: arr = [0 0 0 0 0 1 1]

Output 2: 2

```
import java.io.*;
import java.util.*;
import java.util.Scanner;

public class Main
{
    public static void main(String[] args) {
        int number;
        System.out.println("Enter the integer: ");

        // Create Scanner object
        Scanner s = new Scanner(System.in);

        // Read the next integer from the screen
        number = s.nextInt();

        if((number & 1) == 1) System.out.println("Given number is odd.");
        else System.out.println("Given number is even.");
    }
}
```

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.*;
import java.util.Scanner;

public class Main
```

```

{   public static int countSetBits(int n){
        int count = 0;
        while (n > 0) {
            count += n & 1;
            n >>= 1;
        }
        return count;
    }
    public static void main(String[] args) {
        int number;
        System.out.println("Enter the integer: ");

        // Create Scanner object
        Scanner s = new Scanner(System.in);

        // Read the next integer from the screen
        number = s.nextInt();

        int answer = countSetBits(number);
        System.out.println("The number of set bits in the given number are " +
answer);
    }
}

```

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.*;
import java.util.Scanner;

public class Main

```

```

{ public static int findOddOccuring(int[] arr)
    {
        int xor = 0;
        for (int i: arr) {
            xor = xor ^ i;
        }
        return xor;
    }
    public static void main(String[] args) {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the number of elements you want to store:
");
        //reading the number of elements from the that we want to enter
        n=sc.nextInt();
        //creates an array in the memory of length 10
        int[] array = new int[10];
        System.out.println("Enter the elements of the array: ");
        for(int i=0; i<n; i++)
        {
            //reading array elements from the user
            array[i]=sc.nextInt();
        }
        System.out.println("The odd occurring element is " +
findOddOccuring(array));
    }
}

```