**Rightmost different bit**

Given two numbers **M** and **N**. The task is to find the position of the **rightmost different** bit in the binary representation of numbers.

**Example 1:**

**Input:** M = 11, N = 9

**Output:** 2

**Explanation:** Binary representation of the given

numbers are: 1011 and 1001,

2nd bit from right is different.

**Example 2:**

**Input:** M = 52, N = 4

**Output**: 5

**Explanation**: Binary representation of the given

numbers are: 110100 and 0100,

5th-bit from right is different.

**User Task:**  
The task is to complete the function **posOfRightMostDiffBit**() which takes**two arguments m and n** and **returns** the **position of first different bits in m and n**. If both m and n are the same then return **-1** in this case.

**Expected Time Complexity:** O(max(log m, log n)).  
**Expected Auxiliary Space:** O(1).

**Constraints:**  
0 <= M <= 109  
0 <= N <= 109