```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:44 CST 2018*/

/*Write a function that takes an unsigned integer and returns the number of ’1' bits it has (also known as the Hamming weight).

        For example, the 32-bit integer ’11' has binary representation 00000000000000000000000000001011, so the function should return 3.*/


import java.util.*;


class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

    TreeNode(int x) {
        val = x;
    }
}

public class Test {
    public static void main(String[] args) {

        int num = 2147483648;

        new Solution().reverseBits(num);
    }
}


public class Solution {
    // you need to treat n as an unsigned value
    public int hammingWeight(int n) {

        int result = 0;

        while (n != 0) {

            int temp = n & 1;
            if (temp != 0) {
                result++;
            }
            n >>>= 1;
        }

        return result;
    }
}
```