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

/*Reverse bits of a given 32 bits unsigned integer.

        For example, given input 43261596 (represented in binary as
        00000010100101000001111010011100), return 964176192 (represented in binary
        as 00111001011110000010100101000000).

        Follow up:
        If this function is called many times, how would you optimize it?*/


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 input = 1;

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


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

        int result = 0;

        for (int i = 0; i < 32; i++) {
            result <<= 1;
            result += n & 1;
            n >>= 1;
        }
        return result;
    }
}
```