

```

/*
每个非负整数 N 都有其二进制表示。例如， 5 可以被表示为二进制 "101"，11 可以用二进制 "1011" 表示，依此类推。注意，除 N = 0 外，任何二进制表示中都不含前导零。
二进制的反码表示是将每个 1 改为 0 且每个 0 变为 1。例如，二进制数 "101" 的二进制反码为 "010"。
给定十进制数 N，返回其二进制表示的反码所对应的十进制整数。
示例 1：
输入：5
输出：2
解释：5 的二进制表示为 "101"，其二进制反码为 "010"，也就是十进制中的 2 。
示例 2：
输入：7
输出：0
解释：7 的二进制表示为 "111"，其二进制反码为 "000"，也就是十进制中的 0 。
示例 3：
输入：10
输出：5
解释：10 的二进制表示为 "1010"，其二进制反码为 "0101"，也就是十进制中的 5 。
提示：
0 <= N < 10^9
来源：力扣 (LeetCode)
链接：https://leetcode-cn.com/problems/complement-of-base-10-integer
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/

```

分析:

- 与476题一毛一样,方法不赘述.

方法一:C\_Solution

```

int bitwiseComplement(int num)
{
    const unsigned int arr[32] =
    {
        0x00000001,0x00000002,0x00000004,0x00000008,
        0x00000010,0x00000020,0x00000040,0x00000080,
        0x00000100,0x00000200,0x00000400,0x00000800,
        0x00001000,0x00002000,0x00004000,0x00008000,
        0x00010000,0x00020000,0x00040000,0x00080000,
        0x00100000,0x00200000,0x00400000,0x00800000,
        0x01000000,0x02000000,0x04000000,0x08000000,
        0x10000000,0x20000000,0x40000000,0x80000000
    };
    int ret_val = 0;
    int i      = 0;
    int mask   = -1;
    do
    {
        if(num<0)

```

```

    {
        break;
    }
    if(num==0)
    {
        ret_val = 1;
        break;
    }

    for(i = 31; i >= 0; i--)
    {
        if(num&arr[i])
        {
            break;
        }
        else
        {
            mask ^= arr[i];
        }
    }
    ret_val = num^mask;
}while(0);
return ret_val;
}

```

/\*

执行结果：

通过

[显示详情](#)

执行用时 :4 ms, 在所有 C 提交中击败了77.22%的用户

内存消耗 :6.9 MB, 在所有 C 提交中击败了5.88%的用户

\*/