

```

import Foundation
/// 剑指 Offer 43. 1~n 整数中 1 出现的次数
/// 输入一个整数 n，求1~n这n个整数的十进制表示中1出现的次数。
/// 例如，输入12，1~12这些整数中包含1 的数字有1、10、11和12，1一共出现了5次。
/// 示例 1:
/// 输入: n = 12
/// 输出: 5
class Solution {
    func countDigitOne(_ n: Int) -> Int {
        /// 当前位数，从个位开始
        var digit = 1
        /// 统计结果
        var res = 0
        /// 高位，除去最低位后的数
        var high = n / 10
        /// 当前位，即最低位
        var cur = n % 10
        /// 低位，记录当前位之前的数
        var low = 0
        while high != 0 || cur != 0 {
            if cur == 0 {
                /// 当前位为 0 时，只需将高位乘以当前位数 digit，并加到结果 res 中
                res += high * digit
            } else if cur == 1 {
                /// 当前位为 1 时，将高位乘以当前位数 digit，并加到结果 res 中，
                /// 同时还需加上低位 low 和 1（当前位为 1 的情况）
                res += high * digit + low + 1
            } else {
                /// 当前位大于 1 时，将（高位 + 1）乘以当前位数 digit，并加到结果 res 中
                res += (high + 1) * digit
            }
            /// 更新低位 low
            low += cur * digit
            /// 更新当前位 cur 和高位 high
            cur = high % 10
            high /= 10
            /// 更新当前位数 digit
            digit *= 10
        }
        return res
    }
}

let solution = Solution()
print(solution.countDigitOne(12))
print(solution.countDigitOne(13))

```