

```

import Foundation
/// **剑指 Offer 44. 数字序列中某一位的数字**
/// 数字以0123456789101112131415...的格式序列化到一个字符序列中。
/// 在这个序列中，第5位（**从下标0开始计数**）是5，第13位是1，第19位是4，等等。
///
/// 请写一个函数，求任意第n位对应的数字。
class Solution {
    /// 观察数字出现的方式：
    /// **数字范围**    **位数**    **数字量**    **连起来总位数**
    ///    1~9          1          9          9
    ///    10~99        2          9x10        2x90
    ///    start~end    digit      9 x start    9 x digit x start
    /// 因此先判断第 n 位所在的数字的位数
    func findNthDigit(_ n: Int) -> Int {
        var (_n, count, digit, start): (Int, Int, Int, Int) = (n, 9, 1, 1)
        /// **因此先判断第 n 位所在的数字的位数**
        while _n > count {
            /// 扣除前面多余的位
            _n -= count
            digit += 1
            start *= 10
            count = 9*digit*start
        }
        /// 此时所求的数字的位数为 digit，它是从 start 开始的
        /// **确定所求数位所在的数字**
        let num: Int = start + Int((_n-1)/digit)
        /// **确定所求数位在 num 的哪一位**
        let numStr = String(Array(String(num))[( _n-1) % digit])
        let res: Int = Int(numStr)!
        return res
    }
}

let solution = Solution()
print(solution.findNthDigit(3))
print(solution.findNthDigit(11))

```