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。

示例 1：

输入： `n = 3`

输出： `3`

示例 2：

输入： `n = 11`

输出： `0`

我们通过观察，可以发现以下规律：

123456789	10 11 ... 98 99	100 ... 999	...
9 个数	90个数	900个数	...
9x1位数字	90x2位数字	900x3位数字	...

对于第 n 位对应的数字，我们令这个数字对应的数为 `target`，然后分三步进行。

- 首先找到这个数字对应的数是几位数，用 `digits` 表示；
- 然后确定这个对应的数的数值 `target`；
- 最后确定返回值是 `target` 中的哪个数字。

比如输入的 n 是 `365`：

1. 经过第一步计算我们可以得到第 365 个数字表示的数是三位数，$n = 365 - 9 - 90 \times 2 = 176$，`digtis = 3`。这时 $n = 176$ 表示目标数字是三位数中的第 176 个数字。
2. 我们设目标数字所在的数为 `number`，计算得到 $number = 100 + 176/3 = 158$，`idx` 是目标数字在 `number` 中的索引，如果 `idx = 0`，表示目标数字是 `number` 中的最后一个数字。
3. 根据步骤2，我们可以计算得到 `idx = n % digits = 176 % 3 = 2`，说明目标数字应该是 `number = 158` 中的第二个数字，即输出为 `5`。

```
class Solution {
public:
    int findNthDigit(int n) {
        //确定 n 在几位数中第多个少位置上
```

```
        long base=9;
        int digit=1;
        while(n-base*digit>0)
        {
            n-=base*digit;
            base*=10;
            digit++;
        }
        //根据 digit 得出几位数，及在几位数中的所处位置，计算 idx
        //确定 number 及 idx
        int idx=n%digit;
        if(idx==0) idx=digit;
        long number=1;
        for(int i=1;i<digit;i++)
        {
            number*=10;//根据位数来构造 number 基础大小
        }
        number+=(idx==digit)?n/digit-1:n/digit;
        //根据 number 和 idx 确定
        for(int i=idx;i<digit;i++)
            number/=10;
        return number%10;
    }
};
```