#### 逻辑航线信息学奥赛系列教程

## 1313: 位数问题

# 题目描述

在所有的N位数中,有多少个数中有偶数个数字3?由于结果可能很大,你只需要输出这个答案对12345取余的值。

### 输入

读入一个数N(N≤1000)。

### 输出

输出有多少个数中有偶数个数字3。

### 输入样例

2

## 输出样例

73

#### 解析

当n小于等于1时,即在所有的1位数中,偶数个3的数量为9。

这是为什么呢?因为在所有的个位数中,除了3以外的所有数字(0-9)都可以看成包含0个3,而0则又是偶数,所以符合题意。

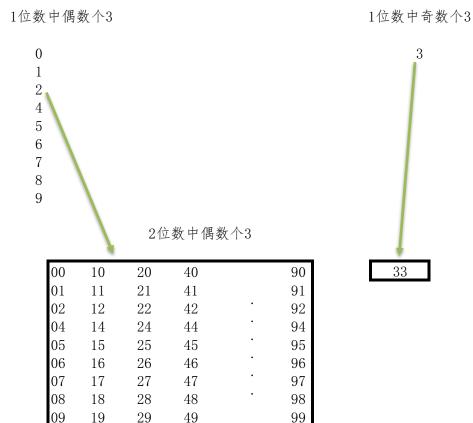
我们用a[1]代表1位数中**偶数个3**的个数,则有a[1]=9。用b[1]代表1位数中**奇数个3**的个数,则有b[1]=1。

n==2时,也就是10--99之间,我们来数一下偶数个3。其中,10--19有9个,20--29有9个,30--39有一个(33),剩下的区间也都是9个。

也就是9+9+1+9+9+9+9+9+9+9=8\*9+1=73个,即a[2]=73,那么奇数个3的个数呢?很明显除了30-39当中有9个,其余的范围中均只有一个,即1+1+9+1+1+1+1+1=17个,即b[2]=17。

那么n=3的时候呢?继续数吗?难道它与n=2或者n=1没有直接的关系吗?

我们当前有一个N位数, N>2。观察下面的图, 我们思考一下这个问题。



通过观察,我们发现两位数中偶数个3的数量分两种情况:

1、来自于1位数中偶数个3的数量,即分别在个位的1-9前加上1,2,4……9,注意没有3。一共是9x9种情况,即,a[2]=a[1] x 9 = 81。

大家比较不好理解的事第一列中以0开头的数字,因为我们的数字是大于2位数的,因此在这里0是可以作为开头数字的,例如100。

其实,通过这一点大家也大概能想明白了,如果当前就是一个两位数,那么则有:  $a[2] = a[1] \times 8 = 72$ 

2、来自于1位数中奇数个3的数量,即在3前再加上3,使其变成33。只有唯一的一种情况。

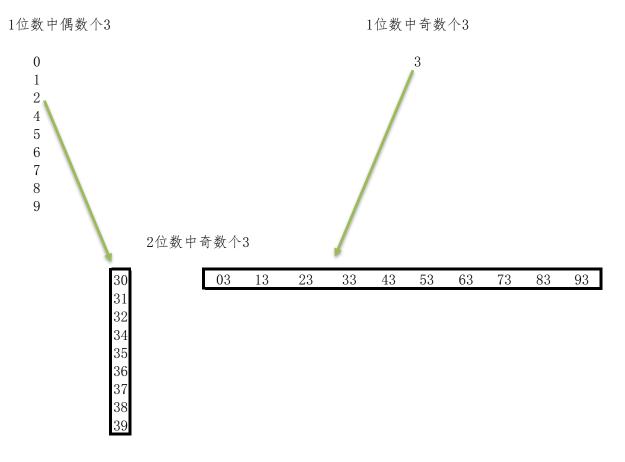
因此, 最终两位数中偶数个3的公式可以写成:  $a[2] = a[1] \times 9 + b[1] \times 1$ 

那么, 我们进一步抽象, 当存在n位数字时, 递推公式可以写成:

$$a[i] = a[i-1] * 9 + b[i] * 1 (i < n)$$

$$a[i] = a[i-1] * 8 + b[i] * 1 (i==n)$$

很明显,上面的公式中还有一个b[i],即在不同的数位下,奇数个3的数量。它是怎么求的呢? 我们继续观察。



相信大家已经看懂了,2位数中奇数个3的来源也有两个: 1、来源于1位数中的偶数个3,即在偶数个3前再加上1个3。公式可以写成:  $b[2] = a[1] \times 1$ 

2、来源于1位数中的奇数个3,即在前面加上0,1,2······9,注意没有3,但是有0。同样的,公式也存在两个:

 $b[2] = b[1] \times 9(非两位数)$  以及  $b[2] = b[1] \times 8(两位数, 排除首位为0的情况)$ 

因此, 最终的递推公式为:

$$b[i] = a[i-1] \times 1 + b[i-1] \times 9 (i \le n)$$

$$b[i] = a[i-1] \times 1 + b[i-1] \times 8 (i==n)$$

# 编码

```
#include <bits/stdc++.h>
using namespace std;
int n, a[1001] = {9}, b[1001] = {1};
int main(int argc, char **argv) {
    //一位数字中的奇偶个3的数量
```

```
a[1] = 9;
   b[1] = 1;
   cin >> n;
   // 一位数字,可以取除了3之外的0-9中任意一个,即0个3
   if (n <= 1) {</pre>
       cout << a[1] << endl;</pre>
   } else {
       int i;
       //i小于n位数的计算
      for (i = 2; i < n; i++) {</pre>
           a[i] = (a[i - 1] * 9 + b[i - 1] * 1) % 12345;
           b[i] = (a[i - 1] * 1 + b[i - 1] * 9) % 12345;
       //i等于n位数的计算
      a[i] = (a[i - 1] * 8 + b[i - 1] * 1) % 12345;
       b[i] = (a[i - 1] * 1 + b[i - 1] * 8) % 12345;
       cout << a[i] << endl;</pre>
   }
   return 0;
}
```

逻辑航线培优教育,信息学奥赛培训专家。

扫码添加作者获取更多内容。

