

E 盗墓笔记之禁婆的考验

题目描述

推开那扇坑爹的石门，忽然一只冷冰冰的手搭在了 Thor 的肩上。回头，一张脸惊得他毛骨悚然——居然是——Arthur？！

长着一张 Arthur 的脸的禁婆露出一个妩媚的微笑。

想过我这一关？先来玩个小游戏吧。很简单，给你一个十进制的数字 n ，你能求出它在 m 进制下的表示里数字 k 的个数么？

于是加法与求模都不会做的 Thor 又逗比了，怎么办？看你喽~

输入

多组测试数据。

每组数据为一行，包含三个整数 n, m, k ($2 \leq m \leq 8$, $0 \leq k \leq 9$)。保证 n 在 `int` 范围内。

输出

对于每组数据，输出一行，包含一个整数，为 k 的个数。

输入样例

9 2 1

8 8 1

输出样例

2

1

Analysis:

本次上机的小 BOSS（喂！为什么没设大 BOSS），AC 人数最少的一个题。首先我们要回想一下进制转换（吕云翔老师可是用中文讲过的！说不知道的去面壁。），比较核心的步骤就是取模（题目中是有提示的）。因为 $m \leq 8$ ，没有寂寞如雪的超过 10 进制，所以难度系数还不是很高。

那么我提一种思路：把这个数字通过进制转换规则转换为 m 进制的数字（其实是 10 进制的数字披上 1 个 m 进制的外衣，各种运算规则本质来说还是 10 进制，只不过我们认为他已经转换为 m 进制了）。转换之后我们每一位模 10 再除以 10，这样就可以从个位开始进行判断是否与 k 相等，再设定一个计数器用来统计次数就 OK 了。是不是思路清晰完整？等等！

以数字 2 为例，我们转换为 2 进制之后是 10，虽然如果是真正的二进制来说的确表示的是 2，但是计算机不是这么认为的， $10-1=9$ 而不是 1，所以一个问题产生了，如果这么做，一个 `int` 范围内的数转成二进制可以有 32 位（想一想，为什么），可以完爆 `long long`，那么窝闷该怎么办。

很简单，抛弃转换完毕才能判断的思想，取模之后立即判断结果是否与 k 相等。

Code for Reference:

```
#include<iostream>
using namespace std;

int main()
{
    int n,m,k,l;
    while(cin>>n>>m>>k)
    {
        l=0;//计数器清零，原来窝闷的小 boss 太年轻…所以代码如此短。
        while(n>0)
        {
            if(n%m==k)
                l++;
            n=n/m;//先取模然后除掉，就酱紫。
        }
        cout<<l<<endl;//这个是输出结果…注释太少不能直视…凑个数。
    }
}

/*只要大家注意到我上面提到的细节就好了，这个题的代码并不复杂*/
```