☑ PAT 乙级题目讲解: 1017《A除以B》

፟ 题目简介

给定两个正整数 A 和 B, 其中:

- *A* 是不超过 **1000 位**的超大正整数;
- B是一位正整数。

要求输出商数 Q 和余数 R, 使得 $A = B \times Q + R$ 成立。

由于 A 过大,无法使用常规整型变量处理,因此需使用**字符串**进行**逐位除法模拟**。

◈ 样例分析

输入样例 1:

123456789050987654321 7

- 模拟除法操作,逐位计算商数:
 - 。 高位不足除以 7, 继续与下一位拼接;
 - 直到被除数大于等于 7, 可计算一位商并更新余数。
 - 。 逐步竖式模拟除法过程:

位次 i	当前位 t	商 q[i]	余数 r
0	1	0	1
1	12	1	5
2	53	7	4
3	44	6	2
4	25	3	4

• 最终商 Q = 17636684150141093474,余数 R = 3。

输出:

17636684150141093474 3

€ 解题思路

本题属于大整数除法模拟问题,整体流程如下:

1. 将超大整数 A 以字符串形式读入, 转成整型数组处理;

- 2. 定义初始余数 r=0;
- 3. 从左至右逐位处理字符型数字,模拟"手算除法"过程:
 - \circ 当前数位参与计算的值为: $r \times 10 + 3 \text{ 前数字}$;
 - \circ 当前位商为: $\left|\frac{r \times 10 + 3 \text{ in } \otimes r}{B}\right|$;
 - 。 更新余数为上式的模;
- 4. 所有位处理完成后,输出拼接得到的商 Q 与最终余数 R。

* 变量说明

变量名	数据类型	含义
a	string	高精度被除数 A
na[]	(int[])	A 拆分后的每一位数字
b	int	除数
t	(int)	当前处理的除数段值
r	int	上一步余数
q[]	(int[])	商的每一位
k	(int)	商的首个非零位索引

☑ Step 1: 读取与预处理输入

- 用字符串读入大整数 A, 用整型变量读入 B;
- 将字符串逐位转为整型数组 na[]。

```
string a;
int na[1005], b, t, r, q[1005];
...
// 1. 把 a -> na[i]
int len = a.size();
for(int i = 0; i < len; i++){
    na[i] = a[i] - '0';
}</pre>
```

☑ Step 2: 模拟除法过程

对 A 的每一位进行以下操作:

- 1. 构造当前被除数段 $t = r \times 10 + na[i]$;
- 2. 计算当前位商: $q[i] = t \div b$;
- 3. 更新余数: $r = t \mod b$ 。

```
// 2. for 逐位模拟计算 -> q[i]
for(int i = 0; i < len; i++){
    t = r * 10 + na[i]; // 当前被除数段
    q[i] = t / b; // 当前位商
    r = t % b; // 余数
}
```

☑ Step 3: 去除前导零并输出

- 从 q[0] 开始, 找到第一个非 0 的位置 k;
- 从 q[k] 到 q[len − 1] 依次输出;
- 最后输出空格和余数 r。

```
// 3. 去前导 0
int k = 0;
while(!q[k] && k < len - 1){
    k++;
}
// 4. 输出
for(int i = k; i < len; i++){
    cout << q[i];
}
cout << " " << r;</pre>
```

☑ 完整代码

```
#include<bits/stdc++.h>
using namespace std;
string a;
int na[1005], b, t, r, q[1005];
int main(){
   cin >> a >> b;
   // 1. 把 a -> na[i]
   int len = a.size();
   for(int i = 0; i < len; i++){}
        na[i] = a[i] - '0';
   // 2. for 逐位模拟计算 -> q[i]
   for(int i = 0; i < len; i++){}
       t = r * 10 + na[i];
       q[i] = t / b;
       r = t \% b;
   // 3. 去前导 0
   int k = 0;
   while(!q[k] \& k < len - 1){
       k++;
    // 4. 输出
    for(int i = k; i < len; i++){
```

```
cout << q[i];
}
cout << " " << r;
return 0;
}</pre>
```

四 常见错误提醒

错误类型	具体表现
用 int 读入 A	大整数溢出,程序崩溃或输出错误
商数前导零未处理	输出以 0001 开头,不符合题目要求
商数拼接逻辑错误	未正确更新每一位 $q[i]$ 或遗漏输出
忽略余数输出	最终输出漏掉余数 R

☑ 总结归纳

፟ 核心方法总结

- 高精度除法的竖式模拟;
- 商数数组逐位构建;
- 余数逐步更新传递。

🗐 技术要点回顾

- 使用字符串 + 数组存储超大整数;
- 手动模拟除法的每一位除、余过程;
- 去除前导 0 的实现技巧。

■ 复杂度分析

时间复杂度: *O*(*n*)空间复杂度: *O*(*n*)

其中n为大整数A的位数(最长 1000 位)。

② 思维拓展

- 扩展到支持小数除法 (输出若干位小数);
- 实现高精度加减乘除统一框架;
- 封装为高精度类支持多种运算符重载。