

# ☑ PAT 甲级题目讲解：1005 《Spell It Right》

## 🧩 题目简介

本题是一道简单模拟题，要求将一个非负整数  $N$  的所有数位相加，最后将得到的 **数位和** 的每一位转为英文单词输出。

- 输入的  $N$  是一个不超过  $10^{100}$  的非负整数；
- 输出要求按英文拼写逐位输出数位和，单词间用空格隔开，末尾不能有多余空格。

## 🔧 样例分析

输入：

12345

分析：

- 数位和为  $1 + 2 + 3 + 4 + 5 = 15$ ；
- 将 15 拆分为两个数字：1 和 5；
- 分别映射为英文单词 "one" 和 "five"。

输出：

one five

## 🔍 解题思路

本题可分为三步：输入处理、求和、英文映射输出。

## 🧠 变量说明

| 变量名                | 类型       | 含义                        |
|--------------------|----------|---------------------------|
| <code>n</code>     | string   | 输入的非负整数（字符串形式）            |
| <code>s</code>     | int      | 所有数位的累加和                  |
| <code>a[]</code>   | int[]    | 将 <code>s</code> 拆解为各个位数字 |
| <code>k</code>     | int      | 数组 <code>a[]</code> 的当前位数 |
| <code>num[]</code> | string[] | 0~9 对应的英文单词数组             |

## ☑ Step 1: 读取字符串并计算所有位之和

由于输入的整数可能达到  $10^{100}$ ，无法用整型变量保存，需用字符串读入。

```
string n;
cin >> n;
int s = 0;
for(int i = 0; i < n.size(); i++){
    s += n[i] - '0'; // 遍历字符串求出各位数字之和;
}
```

## ☑ Step 2: 特殊情况特判：输入为 0

```
if(n == "0"){
    cout << "zero";
    return 0;
}
```

## ☑ Step 3: 将数位和拆解为数字数组

将整数  $s$  拆分为每一位，保存在数组中（从个位开始逆序存储）：

```
int a[100], k = 0;
while(s){
    a[++k] = s % 10;
    s /= 10;
}
```

## ☑ Step 4: 输出对应英文单词

用英文单词数组完成数字到字符串的映射输出，并控制空格：

```
string num[10] = {
    "zero", "one", "two", "three", "four",
    "five", "six", "seven", "eight", "nine"
};

for(int i = k; i >= 1; i--){
    cout << num[a[i]];
    if(i != 1) cout << " "; // 最后一个数后面不加空格
}
```

## ☑ 完整代码

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    string n;
    cin >> n;
    if(n == "0"){
```

```
        cout << "zero";
        return 0;
    }
    int s = 0, len = n.size();
    for(int i = 0; i < len; i++){
        s += n[i] - '0';
    }
    int a[100], k = 0;
    while(s){
        a[++k] = s % 10;
        s /= 10;
    }
    string num[10] = {"zero", "one", "two", "three", "four",
                     "five", "six", "seven", "eight", "nine"};
    for(int i = k; i >= 1; i--){
        cout << num[a[i]];
        if(i != 1) cout << " ";
    }
    return 0;
}
```

## 🚩 常见错误提醒

| 错误类型     | 具体表现                   |
|----------|------------------------|
| 使用整型读入   | 输入数据过大导致溢出             |
| 输出顺序反了   | 数位拆解后未倒序输出             |
| 多余空格     | 输出末尾加空格或中间多空格          |
| 未特判输入为 0 | 输入 "0" 情况下未正确输出 "zero" |

## ✅ 总结归纳

### 🔗 核心方法总结

- 大整数使用字符串读取；
- 遍历字符求数位和；
- 数组映射数字到英文输出。

### 📄 技术要点回顾

- 字符串与整数互转；
- 数位分解与顺序控制；
- 输出格式控制技巧。

### 📊 复杂度分析

- 时间复杂度： $\mathcal{O}(n)$
- 空间复杂度： $\mathcal{O}(n)$

其中  $n$  为输入字符串长度，最大不超过 100。

---

## 思维拓展

---

- 如果要求直接输出英文表达形式如 “fifteen” 呢？
- 支持不同语言映射（如中文数字），应如何设计映射表？
- 若需要输出全称（如 “one hundred twenty-three”），应如何处理语法规则？