

# ☑ PAT 乙级题目讲解：1009 《说反话》

## 🔗 题目简介

本题要求对输入的一句英文句子进行**单词反转输出**。所谓“说反话”，不是反转字符顺序，而是将句子中各个**单词的顺序**进行逆序输出。

输入保证：

- 一行内的字符串长度不超过 80；
- 各单词之间仅用一个空格分隔；
- 不含多余空格。

## 🔗 样例分析

输入：

```
Hello world Here I Come
```

分析过程：

- 原始顺序：Hello → World → Here → I → Come
- 反转顺序：Come → I → Here → World → Hello

因此输出为：

```
Come I Here World Hello
```

## 🔍 解题思路

### 🗣 变量说明

变量名	含义
<code>s[i]</code>	第 <code>i</code> 个单词，存入字符串数组中
<code>k</code>	当前已经读入的单词数量， <code>s[0] ~ s[k-1]</code> 为有效内容

## ☑ Step 1: 逐词读取输入句子

使用 `while(cin >> s[k])` 逐个读取单词，并存入字符串数组 `s[]` 中，同时将 `k++` 以便下次读入。

注意事项：

- 第一个单词存入 `s[0]`，第二个单词存入 `s[1]`，.....
- 由于 `cin >>` 读取单词时遇到空格自动分隔，因此不需手动分割；
- 循环结束时 `k` 表示读入单词的个数；

- 有效单词范围为 `s[0] ~ s[k-1]`。

```
while(cin >> s[k]){
    k++;
}
```

## ☑ Step 2: 逆序输出单词

从 `s[k-1]` 开始逆序输出至 `s[0]`，中间以空格连接。

注意：

- 最后一个单词（即原句第一个单词）后不应有空格；
- 所以仅当 `i > 0` 时输出空格。

```
for(int i = k - 1; i >= 0; i--){
    cout << s[i] << (i > 0 ? " " : "");
}
```

## ☑ 完整代码

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

string s[100];
int k;
int main(){
    while(cin >> s[k]){
        k++; // s[0] ~ s[k-1]
    }
    for(int i = k - 1; i >= 0; i--){
        cout << s[i] << (i > 0 ? " " : "");
    }
    return 0;
}
```

## 🚫 常见错误提醒

错误类型	具体表现
忘记输出空格	所有单词之间要加空格，除了最后一个
输出多余空格	最后一个单词后不能输出空格，应特判 <code>i &gt; 0</code>
理解错下标范围	实际读入的单词为 <code>s[0] ~ s[k-1]</code> ，不是到 <code>s[k]</code>

## ☑ 总结归纳

- `cin >> s[k]` 可自动按空格分词；

- 记录单词数量 `k` 后，逆序输出；
  - 注意边界条件与空格输出逻辑；
  - 本题为基础输入处理与字符串数组操作，适合作为输入输出训练的入门题。
- 

## 思维拓展

---

- 如果输入句子包含标点符号，是否能正确处理？
- 如果输入中有多余空格，如何预处理？
- 本题也可用 `stack` 来完成逆序输出 —— 将每个单词入栈，最后依次弹出。