☑ PAT 乙级题目讲解：1005《继续(3n+1)猜想》

🧩 题目简介

本题延续了 1001 题的“(3n+1)猜想”，但这次输入的是一组正整数，任务是找出这些数中 **“关键数”**，

即：**没有出现在其他数字的验证路径中的数。**

题目要求将所有关键数按**从大到小**的顺序输出。

🩹 样例分析

输入：

```
6
3 5 6 7 8 11
```

逐个分析每个数字的路径：

- 3 → 5 → 8 → 4 → 2 → 1
- 5 → 8 → 4 → 2 → 1
- 6 → 3 → 5 → 8 → 4 → 2 → 1
- 7 → 11 → 17 → 26 → 13 → 20 → 10 → 5 → …
- 8 → 4 → 2 → 1
- 11 → 17 → …

我们发现：

- `6` 与 `7` 的路径中包含了很多其它数字；
- 但 `6` 和 `7` 本身没有出现在其它数字的路径中 → 它们是关键数。

因此输出为：

```
7 6
```

🔍 解题思路

🧬 关键变量说明

变量名	含义
`k`	输入的数字个数
`t`	当前读入并处理的数字
`a[]`	标记哪些数字是输入原始数字
`f[]`	标记哪些数字出现在路径中
`b[]`	存放筛选出的关键数

本题的解决流程可以分为以下几个步骤：

☑ Step 1：读入所有数字，记录原始输入

我们需要读入 k 个正整数，标记每一个原始数字 `a[t] = 1`，方便后续判断其是否为关键数。

```
scanf("%d", &k);
while(k--){
    scanf("%d", &t);
    a[t] = 1; // 标记为原始输入
```

☑ Step 2：对每个输入数字执行卡拉兹猜想路径变换

- 如果是偶数：$t = t/2$
- 如果是奇数：$t = (3 * t + 1)/2$

将整个路径中经过的数字全部标记在数组 `f[]` 中：

```
    while(t != 1){
        if(t % 2 == 0) t /= 2;
        else t = (3 * t + 1) / 2;
        f[t] = 1; // 出现在路径中
    }
```

☑ Step 3：筛选关键数

关键数满足：

- 它是原始输入（`a[i] == 1`）
- 它没有出现在任何路径中（`f[i] == 0`）

我们按照从大到小的顺序枚举并输出这些数：

```
for(int i = 100; i > 1; i--){
    if(a[i] && !f[i]){
        b[++j] = i;
    }
}
```

☑ Step 4：输出格式控制

注意：数字之间用空格隔开，末尾不带空格。

```
for(int i = 1; i <= j; i++){
    printf("%d", b[i]);
    if(i < j) printf(" ");
}
```

完整代码

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

int k, t;
bool f[10005], a[105];

int main(){
    scanf("%d", &k);
    while(k--){
        scanf("%d", &t);
        a[t] = 1; // 标记 t 是数列中待验证数字
        while(t != 1){
            if(t % 2 == 0){
                t /= 2;
            }
            else{
                t = (3 * t + 1) / 2;
            }
            f[t] = 1; // 标记验证路径中出现过的数字
        }
    }

    int b[105]= {}, j = 0;
    for(int i = 100; i > 1; i--){
        if(a[i] && !f[i]){
            b[++j] = i; // 找出所有关键数存到 b[1] ~ b[j]
        }
    }
    for(int i = 1; i <= j; i++){
        printf("%d", b[i]);
        if(i < j) printf(" ");
    }
    return 0;
}
```

常见错误提醒

错误类型	具体表现
输入未标记	忘记 `a[t] = 1`，无法识别原始输入
顺序错误	正确顺序应为从大到小枚举（100 → 1）
输出格式错误	忽略最后一个数字后不能有空格
数组越界	`f[t]` 或 `a[t]` 空间开太小，导致崩溃

☑ 总结归纳

本题是集合判定与路径覆盖思想的结合实践，建议作为 1001 题的进阶练习来理解。

- 熟练掌握 卡拉兹猜想 模拟建模；
- 学会在路径遍历中构建覆盖集合；
- 筛选出未被覆盖的“关键节点”；
- 注重输出格式控制，避免低级失误。

🧠 思维拓展

本题的核心是构造一套“路径逆向验证系统”：关键数必须“独立存在，不被其他路径覆盖”。

本质是集合操作：

- 输入集合 `A`
- 路径覆盖集合 `F`
- 输出集合 = `A - F`
- 设计 `f[]` 与 `a[]` 两套标记数组，分别记录路径与输入集合，是处理集合差集的一种经典思路。