☑ PAT 甲级题目讲解：1002《A+B for Polynomials》

🧩 题目简介

本题要求实现两个一元多项式 $A(x)$ 与 $B(x)$ 的加法操作。

每个多项式使用若干个 **指数-系数** 对表示，最终输出它们之和（结果中的零系数项应省略）。

📏 样例分析

输入样例：

```
2 1 2.4 0 3.2
2 2 1.5 1 0.5
```

表示两个多项式分别为：

- $A(x) = 2.4x^1 + 3.2$
- $B(x) = 1.5x^2 + 0.5x^1$

合并相同指数项后：

- $A(x) + B(x) = 1.5x^2 + (2.4 + 0.5)x^1 + 3.2 = 1.5x^2 + 2.9x^1 + 3.2$

输出：

```
3 2 1.5 1 2.9 0 3.2
```

🔍 解题思路

本题是典型的“稀疏多项式合并”问题，需注意指数相同项的合并与零系数项的剔除。

使用一个长度为 1005 的数组 `b[]`，以指数为下标、系数为值，来存储合并后的多项式。

📌 变量说明

变量名	类型	含义
`k`	`int`	当前读入的非零项个数
`n`	`int`	当前项的指数
`a`	`double`	当前项的系数
`b[]`	`double[]`	存储每个指数对应的系数
`maxn`	`int`	当前出现的最大指数
`c`	`int`	合并后非零项的数量

☑ Step 1：输入并合并多项式项

封装一个 `f()` 函数用于处理每组多项式的输入：

1. 读取 `k` 对 **指数-系数** 对；
2. 每次累加到数组 `b[]` 的对应下标上；

```
void f(){
    scanf("%d", &k);
    while(k--){
        scanf("%d %lf", &n, &a); // 读取 k 对 指数-系数
        maxn = max(n, maxn); // maxn 记录最大指数项值
        b[n] += a; // 指数为 n 的项系数加上 a
    }
}
```

调用两次 `f()` 函数分别读入 $A(x)$ 与 $B(x)$：

```
f();  // 读 A
f();  // 读 B
```

☑ Step 2：统计非零项个数

遍历 `b[0..maxn]`，判断是否为非零项（浮点数判断），统计项数 `c`：

```
for(int i = 0; i <= maxn; i++){
    if(b[i]) c++;
}
```

☑ Step 3：按降幂输出结果

1. 输出总项数 `c`；
2. 再从 `maxn` 降序到 `0` 输出所有非零项；
3. 每项格式为 `指数 系数`，系数保留 1 位小数。

```
printf("%d", c);
for(int i = maxn; i >= 0; i--){
    if(b[i]){
        printf(" %d %.1lf", i, b[i]);
    }
}
```

☑ 完整代码

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

```
int k, n, maxn, c;
double a, b[1005];

void f(){
    scanf("%d", &k);
    while(k--){
        scanf("%d %lf", &n, &a);
        maxn = max(n, maxn);
        b[n] += a;
    }
}

int main(){
    f();
    f();
    for(int i = 0; i <= maxn; i++){
        if(b[i]) c++;
    }
    printf("%d", c);
    for(int i = maxn; i >= 0; i--){
        if(b[i]){
            printf(" %d %.1lf", i, b[i]);
        }
    }
    return 0;
}
```

常见错误提醒

错误类型	具体表现
忽略合并同类项	相同指数项未累加，导致重复项
浮点数精度丢失	未保留一位小数或误差未处理
忘记剔除零系数项	输出了系数为 0 的项
输出格式不规范	多余空格、换行或小数位数错误

总结归纳

核心方法总结

- 使用数组下标作为指数；
- 直接累加系数；
- 排序输出 + 精度控制。

技术要点回顾

- 多项式的稀疏表示；
- 浮点加法的误差控制；
- 格式化输出技巧（保留小数、控制空格）。

复杂度分析

- 时间复杂度：$\mathcal{O}(n)$
- 空间复杂度：$\mathcal{O}(1001)$

其中 n 为两多项式项数之和，最大指数不超过 1000。

思维拓展

- 多项式乘法该如何实现？能否用 map 优化？
- 若存在负指数或小数指数，该如何表示和合并？
- 若输入项数 $> 10^5$，该如何减少空间消耗？