# ☑ PAT 乙级题目讲解：1010《一元多项式求导》

## 🧩 题目简介

本题考查的是一元多项式求导法则的程序实现。给出一个按“系数 指数”顺序输入的多项式表示，需输出它的一阶导数形式。

导数计算规则：
若某一项为 $ax^b$，则其一阶导数为 $abx^{b-1}$。

但需注意：

- 常数项（指数为 0）导数为 0，应忽略；
- 若导数结果为空（即输入为常数多项式），则应输出 `0 0`；
- 输出为“**系数 指数**”格式，项之间用空格分隔，**末尾不得多空格。**

---

## 🧪 样例分析

输入：

```
3 4 -5 2 6 1 -2 0
```

输入多项式为：

$$3x^4 - 5x^2 + 6x - 2$$

逐项求导：

- $3x^4 \Rightarrow 12x^3$
- $-5x^2 \Rightarrow -10x$
- $6x \Rightarrow 6$
- $-2$ 为常数项，其导数为 0，忽略

输出为：

```
12 3 -10 1 6 0
```

---

## 🔍 解题思路

### 📌 变量说明

| 变量名 | 含义 |
| --- | --- |
| `a` | 当前项的系数 |
| `b` | 当前项的指数 |
| `c[]` | 存放导数结果的数组（系数与指数交替存放） |
| `k` | 当前存入 `c[]` 的索引指针 |
| `f` | 标记是否至少存在一个有效导数项，用于特判零多项式 |

## ☑ Step 1：逐对读取多项式项

通过 `while(cin >> a >> b)` 实现成对读取输入。

```
while(cin >> a >> b){
    ...
}
```

## ☑ Step 2：跳过常数项

若指数为 0，该项为常数，导数为 0，应跳过。

```
if(!b) continue;
```

## ☑ Step 3：计算导数并存入数组

按导数法则求导并存入数组中备用。

```
c[++k] = a * b; // 新系数
c[++k] = b - 1; // 新指数
f = 1;          // 标记至少存在一项
```

## ☑ Step 4：特判零多项式

若 `f == 0`，即没有一项导数有效，应直接输出：

```
0 0
```

## ☑ Step 5：格式化输出结果

遍历 `c[]`，注意末尾无多余空格。

```
for(int i = 1; i <= k; i++){
    cout << c[i] << (i < k ? " " : "");
}
```

---

# ☑ 完整代码

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

int a, b, c[2050], k;
int main(){
    bool f = 0;
    while(cin >> a >> b){
        if(!b) continue; // 跳过常数项计算
        c[++k] = a * b; // 导数项的系数
        c[++k] = b - 1; // 导数项的指数
        f = 1;
    }
    if(!f) cout << "0 0";
    for(int i = 1; i <= k; i++){
        cout << c[i] << (i < k ? " " : "");
    }
    return 0;
}
```

## 常见错误提醒

| 错误类型 | 具体表现 |
| --- | --- |
| 忘记跳过常数项 | 指数为 0 的项仍被处理，导致错误 |
| 输出格式错误 | 多输出空格或结尾处空格 |
| 忽略“零多项式”特判 | 所有项为常数时未输出 `0 0` |

## 总结归纳

- 熟悉一元多项式导数规则；
- 注意特殊情况处理（常数项、零多项式）；
- 精确控制输出格式；
- 简单数组模拟处理足矣，无需复杂结构。

## 时间复杂度

- **时间复杂度：** $O(n)$
  每项读取和处理一次，线性复杂度。
- **空间复杂度：** $O(n)$
  使用一维数组记录导数结果，空间线性增长。

## 思维拓展

- 若输入项很多，考虑链表/向量优化；
- 多项式乘法、积分等操作也可参考类似结构；
- 本题是表达式建模与结构化存储的启蒙案例之一。

> 本题虽然是基础题，但细节众多，需要精细操作与完整逻辑链。能正确写出并通过所有测试点，说明你对表达式建模和边界处理已初步掌握！