

☑ PAT 甲级题目讲解：1009 《Product of Polynomials》

🧩 题目简介

本题要求实现两个多项式的乘法。

即给定两个稀疏多项式 $A(x)$ 和 $B(x)$ ，求出它们的乘积多项式 $C(x) = A(x) \times B(x)$ 。

最后输出非零项（指数从大到小排列，保留一位小数）。

🔧 样例分析

输入样例：

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

两个多项式分别为：

- $A(x) = 2.4x^1 + 3.2x^0$
- $B(x) = 1.5x^2 + 0.5x^1$

手动展开乘积：

$$\begin{aligned}C(x) &= (2.4x^1 + 3.2) \times (1.5x^2 + 0.5x^1) \\&= 2.4x^1 \times 1.5x^2 + 2.4x^1 \times 0.5x^1 + 3.2x^0 \times 1.5x^2 + 3.2x^0 \times 0.5x^1 \\&= 3.6x^3 + 1.2x^2 + 4.8x^2 + 1.6x^1\end{aligned}$$

合并同类项后：

- $C(x) = 3.6x^3 + 6.0x^2 + 1.6x^1$

输出为：

```
3 3 3.6 2 6.0 1 1.6
```

🔍 解题思路

本题的本质是稀疏多项式的乘法展开 + 合并同类项 + 格式控制输出。

多项式乘法的核心是逐项相乘：

$$A(x) = \sum_{i=1}^{k_1} a_i x^{n_i}, \quad B(x) = \sum_{j=1}^{k_2} b_j x^{m_j}$$

则：

$$C(x) = A(x) \times B(x) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} a_i b_j x^{n_i + m_j}$$

变量说明表格

变量名	含义
<code>k1, k2</code>	多项式 A 和 B 的项数
<code>n[i], a[i]</code>	多项式 A 的第 i 项：指数和系数
<code>m[i], b[i]</code>	多项式 B 的第 i 项：指数和系数
<code>c[i]</code>	指数为 i 的项的系数（乘积结果）
<code>maxn</code>	当前记录的最大指数值
<code>s</code>	非零项的个数

☒ Step 1: 输入两个稀疏多项式

```
cin >> k1;
for(int i = 1; i <= k1; i++){
    cin >> n[i] >> a[i]; // A 的指数和系数
}
cin >> k2;
for(int i = 1; i <= k2; i++){
    cin >> m[i] >> b[i]; // B 的指数和系数
}
```

☒ Step 2: 逐项相乘，累加结果

模拟 $C(x) = A(x) \times B(x)$ 的计算过程，数学公式为：

$$C(x) = A(x) \times B(x) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} a_i b_j x^{n_i + m_j}$$

代码实现使用双层循环逐项相乘，累加结果：

```
for(int i = 1; i <= k1; i++){ // A 有 k1 项
    for(int j = 1; j <= k2; j++){ // B 有 k2 项
        int t = n[i] + m[j]; // 新项指数
        c[t] += a[i] * b[j]; // 合并同类项
        maxn = max(maxn, t); // 更新最大指数
    }
}
```

☒ Step 3: 统计非零项数 + 按指数从大到小输出

```

for(int i = 0; i <= maxn; i++){
    if(c[i]) s++; // 非零项统计
}
printf("%d", s);
for(int i = maxn; i >= 0; i--){
    if(c[i]){ // 该项非 0
        printf(" %d %.11f", i, c[i]); // 注意先输出空格
    }
}
}

```

☑ 完整代码

```

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

int k1, k2, n[15], m[15], maxn, s;
double a[15], b[15], c[2005];

int main(){
    cin >> k1;
    for(int i = 1; i <= k1; i++){
        cin >> n[i] >> a[i];
    }
    cin >> k2;
    for(int i = 1; i <= k2; i++){
        cin >> m[i] >> b[i];
    }
    for(int i = 1; i <= k1; i++){
        for(int j = 1; j <= k2; j++){
            int t = n[i] + m[j];
            c[t] += a[i] * b[j];
            maxn = max(maxn, t);
        }
    }
    for(int i = 0; i <= maxn; i++){
        if(c[i]) s++;
    }
    printf("%d", s);
    for(int i = maxn; i >= 0; i--){
        if(c[i]){
            printf(" %d %.11f", i, c[i]);
        }
    }
    return 0;
}

```

🚫 常见错误提醒

错误类型	错误描述
小数精度不对	忘记使用 <code>%.11f</code> 输出浮点数
忘记或多输出空格	空格输出要严格控制
忘记合并同类项	没有用 <code>+=</code> ，会导致覆盖而非累加

☑ 总结归纳

- 本题为**稀疏多项式乘法模拟**；
- 建议使用数组作为指数映射表 `c[i]`，空间换时间；
- 注意控制：
 - 合并同类项；
 - 按指数从大到小输出；
 - 浮点数格式 `%.11f`；
 - 严格空格输出。

时间复杂度约为 $O(k_1 \times k_2)$ ，完全可接受。

🧠 思维拓展

- 如果改成多项式求和，只需将对应 `c[i]` 做加法；
- 更复杂情形可引入结构体表示项，并排序；
- 这类题是数据结构课程中“稀疏数组”模型的重要应用。