

☑ PAT 甲级题目讲解：1001 《A+B Format》

🧩 题目简介

本题要求你计算两个整数 a 和 b 的和，并以**英文标准格式**输出结果：

- 数字从右向左每三位添加一个逗号 `,`；
- 若不足四位则不加逗号；
- 注意处理负数时负号应保留在首位。

🔧 样例分析

输入样例：

```
-1000000 9
```

计算结果为：

```
-999991
```

添加千位分隔符后输出为：

```
-999,991
```

🔍 解题思路

本题核心是对计算结果进行字符串格式化处理。解法共分四步：

🧠 变量说明

变量名	类型	含义
<code>a</code>	<code>int</code>	输入的第一个整数
<code>b</code>	<code>int</code>	输入的第二个整数
<code>c</code>	<code>int</code>	两数之和
<code>d[]</code>	<code>int[]</code>	拆分后的每一位数字（逆序）
<code>k</code>	<code>int</code>	数字位数
<code>s</code>	<code>string</code>	拼接后的带逗号字符串

☑ Step 1: 读取输入并计算和

读取 a, b , 计算其和 c 。

```
int a, b, c;
cin >> a >> b;
c = a + b;
```

若 $|c| < 1000$, 直接输出结果, 无需格式化:

```
if(abs(c) < 1000){
    cout << c;
    return 0;
}
```

☑ Step 2: 处理符号与拆分数位

若 $c < 0$, 先输出负号, 并转为正数:

```
if(c < 0){
    cout << "-";
    c = -c;
}
```

进行数位拆分, 逆序存入数组:

```
int d[15], k = 0;
while(c){
    d[++k] = c % 10;
    c /= 10;
}
```

☑ Step 3: 构造带千位分隔符的字符串

从低位到高位逐位拼接, 并在每 3 位添加 `,` (除最高位):

```
string s = "";
for(int i = 1; i <= k; i++){
    s += d[i] + '0';
    if(i % 3 == 0 && i != k){ // 注意特判最高位 i != k
        s += ',';
    }
}
```

☑ Step 4: 反转字符串并输出

最终字符串是反序的, 需翻转后输出:

```
reverse(s.begin(), s.end());
cout << s;
```

☑ 完整代码

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

int main(){
    int a, b, c;
    cin >> a >> b;
    c = a + b;

    if(abs(c) < 1000){
        cout << c;
        return 0;
    }

    if(c < 0){
        cout << "-";
        c = -c;
    }

    int d[15], k = 0;
    while(c){
        d[++k] = c % 10;
        c /= 10;
    }

    string s = "";
    for(int i = 1; i <= k; i++){
        s += d[i] + '0';
        if(i % 3 == 0 && i != k){
            s += ',';
        }
    }
    reverse(s.begin(), s.end());
    cout << s;
    return 0;
}
```

🚫 常见错误提醒

错误类型	具体表现
忽略负数符号输出	未处理 <code>-</code> 号或位置错误
分隔逗号位置错误	没有每 3 位加，或加在最高位
忘记翻转拼接结果	直接输出拼接结果，顺序颠倒
忽略 0 及小于 1000 的特判	答案 0 及其它绝对值小于 1000 的数都直接输出就行

☑ 总结归纳

🔑 核心方法总结

- 先求和，判断是否需要格式化；
- 拆分数位并添加逗号；
- 注意符号处理与字符串翻转。

📄 技术要点回顾

- 数位拆分模板；
- 字符拼接技巧；
- STL 函数 `reverse()` 应用。

🏠 复杂度分析

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

其中 n 为结果整数的位数（最多 7~10 位）。

🧠 思维拓展

- 若结果为浮点数如何保留小数格式化？
- 是否可将格式化逻辑封装为函数供复用？
- C++ 中是否有内置格式化方法可替代该模拟？