

☑ PAT 乙级题目讲解：1011 《A+B 和 C》

🔗 题目简介

本题要求判断多个测试用例中，是否满足：

$$A + B > C$$

其中 A、B、C 是给定范围内的整数，重点在于格式化输出以及 **避免整型溢出** 问题。

🔧 样例分析

输入：

```
4
1 2 3
2 3 4
2147483647 0 2147483646
0 -2147483648 -2147483647
```

逐组分析：

- Case 1: $1 + 2 = 3$ ，不大于 3 → **false**
- Case 2: $2 + 3 = 5$ ，大于 4 → **true**
- Case 3: $2147483647 + 0 = 2147483647 > 2147483646$ → **true**
- Case 4: $0 + (-2147483648) = -2147483648 < -2147483647$ → **false**

输出：

```
Case #1: false
Case #2: true
Case #3: true
Case #4: false
```

🔍 解题思路

🧠 变量说明

变量名	含义
t	测试用例组数
a	当前组的第一个整数
b	当前组的第二个整数
c	当前组的第三个整数（比较对象）
i	当前测试用例编号，从 1 开始

本题的解法非常直接，但**务必注意两个点**：

☑ Step 1: 使用 `long long` 防止溢出

虽然题目中 A、B、C 看上去都在 `int` 范围内，但 A+B 可能超过 `int` 的最大值（如样例中的第 3 组），因此必须使用 `long long` 存储：

```
long long a, b, c;
```

☑ Step 2: 逐组判断并输出格式

输出格式要求为：

```
Case #编号: true/false
```

每组编号从 1 开始。

使用 `cout << "Case #" << i << ": "` 控制格式：

```
for(int i = 1; i <= t; i++){
    cin >> a >> b >> c;
    cout << "Case #" << i << ": "; // printf("Case #%d: ", i);
    if(a + b > c){
        cout << "true";
    } else {
        cout << "false";
    }
    if(i < t) cout << "\n";
}
```

☑ 完整代码

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

long long a, b, c;
int t;
int main(){
    cin >> t;
    for(int i = 1; i <= t; i++){
        cin >> a >> b >> c;
        cout << "Case #" << i << ": ";
        if(a + b > c){
            cout << "true";
        }
        else{
            cout << "false";
        }
        if(i < t) cout << "\n";
    }
    return 0;
}
```

🚩 常见错误提醒

错误类型	具体表现
忽略溢出问题	用 int 存储 A+B, 可能导致错误判断
Case 编号错误	测试编号从 0 开始或不输出编号
输出格式错误	Case #X: 后面空格缺失, 大小写错误, 或多空格
多余换行	每组输出后加多余换行, 或漏掉换行

✅ 总结归纳

- 本题核心是数据边界判断和格式输出控制;
- 注意使用 long long 避免溢出问题;
- 输出格式必须严格按照 Case #X: true/false 编写;
 - cout 写法: `cout << "Case #" << i << ": "`;
 - printf() 写法: `printf("Case #%d: ", i);`
- 时间复杂度: $O(T)$
- 空间复杂度: $O(1)$

🧠 思维拓展

- 若将题目拓展为处理浮点数相加, 如何考虑精度误差?
- 若是要判断 $(A + B > C)$ 但不能用 + 运算符, 该如何设计? 可引入更底层的移位运算等方式。