

# ☑ PAT 甲级题目讲解：1011 《World Cup Betting》

## 🧩 题目简介

本题以 2010 南非世界杯为背景，讲述了中国体育彩票提供的一个叫做 “Triple Winning（三选一）” 的投注玩法。

每场比赛都有三个可能的结果可以投注：

- **W** (Win) : 胜
- **T** (Tie) : 平
- **L** (Lose) : 负

每个结果都有对应赔率，即押中后按赔率收益。

玩家选择任意 **3 场比赛**，每场选择一个结果投注，**总收益计算方式**为：

$$\text{profit} = (a_1 \times a_2 \times a_3 \times 0.65 - 1) \times 2$$

- $a_1, a_2, a_3$  : 三场比赛中各自选择的最大赔率；
- **0.65** 是系统扣率（即乘积赔率会乘上 65%）；
- 减去 **1** 是减掉本金；
- 再乘 **2** 是因为下注金额为 2 元。

要求获得最大收益的下注顺序以及最大收益是多少。

## 🔍 样例分析

输入样例：

```
1.1 2.5 1.7
1.2 3.1 1.6
4.1 1.2 1.1
```

分析过程：

- 第一轮最大赔率 2.5 (**T**) ；
- 第二轮最大赔率 3.1 (**T**) ；
- 第三轮最大赔率 4.1 (**W**) ；

最大收益计算如下：

$$\begin{aligned}\text{ans} &= (2.5 \times 3.1 \times 4.1 \times 0.65 - 1) \times 2 \\ &= (31.775 \times 0.65 - 1) \times 2 \\ &= (20.65375 - 1) \times 2 = 19.65375 \times 2 = 39.3075 \approx 39.31\end{aligned}$$

输出下注顺序，然后保留两位小数输出最大收益：

## 🔍 解题思路

考察基本的 **选择+乘法+浮点运算+字符串映射输出** 的能力。解题核心是 **每轮选择最大赔率**，并记录**对应下注选项**，最后进行浮点乘法与格式化输出。

## 📝 变量说明表格

变量名	类型	含义
c[]	char[]	映射下标到选项字符 'W', 'T', 'L'
a[]	double[]	当前一轮三种赔率
ans	double	最终的累计收益乘积
maxx	double	当前一轮最大赔率值
d	int	当前一轮最大赔率对应选项下标 (1~3)

## ☑ Step 1: 初始化选项映射与累计收益

```
char c[5] = {' ', 'W', 'T', 'L'};  
double a[5], ans = 1;
```

## ☑ Step 2: 封装处理每一轮赔率的函数

使用函数 f() 来：

- 读取一轮的三个赔率；
- 找出最大值及其位置；
- 累乘最大赔率到 ans；
- 输出对应的选项字符和空格。

```
void f(){  
    double maxx = 0;  
    int d = 0;  
    for(int i = 1; i <= 3; i++){  
        scanf("%lf", &a[i]);  
        if(a[i] > maxx){  
            maxx = a[i];  
            d = i;  
        }  
    }  
    ans *= maxx;  
    printf("%c ", c[d]);  
}
```

### ☑ Step 3: 主函数执行流程

1. 初始化收益乘积 `ans = 1`;
2. 调用三次 `f()`，对应三轮比赛；
3. 按题目规则计算最终收益；
4. 输出保留两位小数的最终收益。

```
int main(){
    f();
    f();
    f();
    ans = (ans * 0.65 - 1) * 2;
    printf("%.2lf", ans);
    return 0;
}
```

### ☑ 完整代码 (C++)

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

char c[5] = {' ', 'W', 'T', 'L'};
double a[5], ans = 1;

void f(){
    double maxx = 0;
    int d = 0;
    for(int i = 1; i <= 3; i++){
        scanf("%lf", &a[i]);
        if(a[i] > maxx){
            maxx = a[i];
            d = i;
        }
    }
    ans *= maxx;
    printf("%c ", c[d]);
}

int main(){
    f();
    f();
    f();
    ans = (ans * 0.65 - 1) * 2;
    printf("%.2lf", ans);
    return 0;
}
```

### 🚫 常见错误提醒

错误类型	具体表现
索引从 0 开始错误	<code>c[]</code> 数组下标与赔率顺序不一致，导致输出选项错误
浮点精度问题	没有使用 <code>%.21f</code> 格式输出，保留位数不对
未初始化收益乘积	忘记 <code>ans = 1</code> ，导致最终收益错误
多余换行或空格	输出格式不符合要求，应仅用空格分隔

## ☑ 总结归纳

### 🔑 核心方法总结

- 每轮找最大值及其位置；
- 映射下标输出对应选项；
- 累乘赔率，按题意公式计算收益。

### 📄 技术要点回顾

- 输入输出格式控制；
- 基础数组操作与映射逻辑；
- 简单浮点乘法与保留位数输出。

### 🏠 复杂度分析

- 时间复杂度： $\mathcal{O}(1)$ （仅处理常数个数据）
- 空间复杂度： $\mathcal{O}(1)$

## 🧠 思维拓展

- 若比赛轮数变为  $n$  轮，如何改写程序支持动态输入？
- 本题实质是：**选择最大收益的策略问题**，可类比：投资选择、博弈策略优化等模型。