

F 盗墓笔记之逃亡

题目描述

电闪雷鸣，警笛大作——不明人士的举报使得乘着列车去往长白山的 Thor 与天真不得不搞了一次夜半逃亡之旅。给你一上帝视角，将他们跑路的路线与警方的追查路线简化为一平面直角坐标系上的两条直线，由于警方人手的优越性，只要 Thor 与天真有可能出现在对方的路线上，就会被包围逮捕。他们只能找一条完全与警方没有交集的路线，才可以安全逃离。现请给你这两条路线的一般表示 ($Ax+By+C=0$)，请你判断他们是否会遭到逮捕。

输入

多组测试数据。

每组测试数据分为两行。第一行包含三个整数 A_1, B_1, C_1 ，用空格隔开，为 Thor 和天真的逃跑路线。第二行包含三个整数 A_2, B_2, C_2 ，用空格隔开，为警方的追捕路线。

保证输入数据可以构成直线。

输出

对于每组数据，输出一行。如果 Thor 与天真会被逮捕，输出“Dead End”。如果他们安全逃离，则输出“Safe and Sound”

输入样例

```
1 1 1
1 1 2
1 1 1
1 2 1
```

输出样例

```
Safe and Sound
Dead End
```

解题分析：

本题考查 if 和 else 的使用，真实的中等水题。抽象模型已经陈述得很明确，就是判断两条直线的位置关系，平行时 OK，不平行时 KO。

几个坑，首先直线重合是算在 KO 里的，其次，ABC 三个数不一定是互质的。（1 2 3 和 2 4 6 其实是一条直线的哦）

然而直线平行的判断条件有通解的，就是 $A_1*B_2 == A_2*B_1$ 且 $A_1*C_2 != A_2*C_1$ ，具体证明略。

（此外，不妨试着用一用 $?$ ：运算符，很有趣的哦）

参考代码:

```
#include<iostream>
#include<fstream>
using namespace std;

bool judge(int a1,int b1,int c1,int a2,int b2,int c2)
{
    return(a1*b2==a2*b1)&&(a1*c2!=a2*c1); //两个判断条件,参见上面的分析
}

int main()
{
    int a1,b1,c1,a2,b2,c2;
    while(cin>>a1>>b1>>c1>>a2>>b2>>c2)
        judge(a1,b1,c1,a2,b2,c2)?
            cout<<"Safe and Sound"<<endl:cout<<"Dead End"<<endl;
}
```