**董与黑的碰撞-侵入**

时间限制：1000MS

**描述：**

大战将至，联盟的机器却相继崩溃，主管数据库的轩神经过反复检查，发现这是黑暗势力远程侵入了系统，篡改了程序，导致计算机中出现了一种叫做死锁的故障。

死锁的原理是这样的：

考虑这样一种情况：在数据库中，多个进程会并发执行，当一个进程希望访问一个数据时，会先给这个数据上锁，直到这个进程结束（提交修改），这个锁才会被解开。这样，其他访问该数据库的进程需要访问该数据时，需要等待锁解开时才可进行访问。例如T1进程先访问了A数据，T2进程在之后想要访问A数据，则需要等待T1进程提交修改。我们称这为一次等待，记做T2 W T1。

由于给数据加锁这一操作在访问数据前一刻进行，可能会出现这种情况：T1 访问数据A，T2访问数据B，T1希望访问数据B， T2希望访问数据A，此时B被T2上锁，A被T1上锁，两个进程均无法结束，两个进程均进入等待，死锁发生。这种情况可以表示为T2 W T1,T1 W T2。同样，在多个进程并发执行时，若出现冲突出现闭环（eg. T1 W T2, T2 W T3, T3 W T1），也会出现死锁。

由于轩神忙于准备战斗，所以他请你帮忙解决死锁的问题。现在告诉你一台机器中的n个进程（编号1,2,3...n)与m个冲突信息，请你判断这台机器是否存在死锁。

**输入：**

首先输入一个正整数T，表示测试数据的组数。

对于每组数据，第一行输入两个数n,m，接下来m行，每行输入两个整数a,b表示(a W b)，保证一组数据中相同的(a,b)不会出现两次

**数据范围：**

T≤500

n≤1000

m≤2000

1≤a,b≤n，且a≠b

**输出：**

对于每组测试数据，若不存在死锁，则输出一行“No Deadlock”，若存在死锁，则输出一行“Deadlock Occurs”。

**样例输入：**

3

3 2

1 2

2 3

2 2

1 2

2 1

4 4

1 2

3 4

1 3

4 1

**样例输出：**

No Deadlock

Deadlock Occurs

Deadlock Occurs