# J - 肆虐的病毒

##### Time Limit: 12000/4000MS (Java/Others)     Memory Limit: 65535/65535KB (Java/Others)

Submit Status

最近你收到一条消息



你发现，你的电脑中了最近到处肆虐的一种病毒。这种病毒甚至通过一种奇怪的方法，改变了你最近晚上的入睡时间。 这种操作使你感到非常的难受，所以你向黑客发了封邮件，恳请他放你一马。

黑客告诉你，他确实可以放你一马。但是，你得和他玩一个游戏，成败全在你自己。

在这个游戏中，有以1⋯n1⋯n编号的，共计nn个物体被放置在一个圆环里面（顺时针）。编号为11的物体是你的女朋友。其他的都是小绿帽。游戏开始的时候，一个小机器人会被随机的放在这nn个物体的其中的一个旁边（放心，当然不会是你的女朋友旁边）。

你和黑客手上都分别有一个数集（里面有从11到n−1n−1的一些数字）。黑客的集合是s1s1,大小是k1k1，你的集合是s2s2，大小是k2k2。你和黑客中的一个人会走先手，之后轮流进行操作。在每个回合中，当前的玩家的意识会和小机器人进行对接，当前的的玩家可以从他的集合中任意的选择一个数xx，然后这个数可以使小机器人沿着顺时针方向走xx步。如果当前玩家操纵小机器人到达了你的女朋友处（11处），那么他就赢得了游戏。每走到一个不是你女朋友的位置，你就会获得一顶小绿帽。

你现在很方，你想知道对于每一种可能的开局（谁先走，机器人放在哪个位置），你的输赢情况（假设黑客和你都绝顶（秃）聪明），于是你打算编写一个程序来解决你的疑惑。要注意，有一种谁都不赢谁的方法，但是这种情况下游戏会无限的循环，这意味着你会得到无限顶小绿帽。

## Input

第一行有一个数字nn（2≤n≤70002≤n≤7000），表示游戏中物体的数量。

第二行有一个k1k1，然后紧接着有k1k1个数。a1,a2⋯aka1,a2⋯ak

第二行有一个k2k2，然后紧接着有k2k2个数。b1,b2⋯bkb1,b2⋯bk

q≤k1，k2≤n−1q≤k1，k2≤n−1 并且 aiai和bibi在11到n−1n−1的范围内。

## Output

第一行输出n−1n−1个单词，用空格隔开。第ii个单词为"Win","Lose"或"Loop"，分别表示黑客先走，机器人一开始在i+1i+1个位置的时候，黑客赢了，输了，或游戏陷入循环。

同样的，第二行输出n−1n−1个单词，用空格隔开。第ii个单词为"Win","Lose"或"Loop"，分别表示你先走，机器人一开始在i+1i+1个位置的时候，你赢了，输了，或游戏陷入循环

## Sample input and output

| **Sample Input** | **Sample Output** |
| --- | --- |
| 5  2 3 2  3 1 2 3 | Lose Win Win Loop  Loop Win Win Win |
| 8  4 6 2 3 4  2 3 6 | Win Win Win Win Win Win Win  Lose Win Lose Lose Win Lose Lose |

## Hint

ac之后还是绿色的，这病毒真是可怕啊。

题意：

一个环，共n个位置，两个人每次可以走的步数是各自的数集中的任意一个数，两人轮流走，走到1号就赢得游戏。问从环上2~n号分别出发，各自先手时的所有情况中，是必胜、必败还是循环游戏，无法有结果。假设两人都采取最优策略。

题解：

博弈+DP。

如果我们枚举出发点，一个个向后推，不仅不好确定何时必胜、何时必败，还可能超时，并产生大量的重复计算。此时，我们不妨从终点向前递推。

首先，如果任意一个人先手，并可以一步到达在一号位置，则在此位置此人必胜。于是，我们可以设立1号点为B的“必败点”，并假设到达这点的人是A，从1号点向前按照A可以走的步数逆推，所得到的点都是A的必胜点。以此类推，所有从B的必败点可以向前推得的点，也就是正着走可以到达任意B的必败点的点，都是A的必胜点。

那么，何时从一个点出发是必败的呢？1号点B必败，这点毋庸置疑。对于其他点呢？我们发现：要想必败，则B从此点出发的所有情况都必须失败，而B可走的步数就是数集中元素的个数。于是，我们可以采用以下算法：

设立一个队列，队列中元素存储此时向前逆推到达的位置和上一步是谁走的，且队列中的位置都已确定胜负关系。初始时，将1号点作为A、B的必胜点分别放入队列，并存储必胜、必败状态。接着，我们反复取队尾元素做如下操作：如果队尾是A的必胜点，则向前逆推可能到达该点的点，并统计到达该点的A的必胜点的个数，如果这个数与A的数集大小相同，则这个点是B的必败点，存储这一状态并将这一位置加入队首。如果队尾是A的必败点，则向前逆推所有可能到达该点的点，将它们更新为B的必胜点，并将它们加入队首。对于B的必胜点和必败点，推理过程相同，将B和A互换即可。重复上述操作，直到队列为空，没有推得必胜、必败状态的点则是循环点。

如此即可利用DP的方法解出这道博弈题。