**Q. Win in a Board Game**

In a board game, a pXp board is given and the cells of the board are numbered as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| 3,0 | 3,1 | 3,2 | 3,3 |
| 2,0 | 2,1 | 2,2 | 2,3 |
| 1,0 | 1,1 | 1,2 | 1,3 |
| 0,0 | 0,1 | 0,2 | 0,3 |

The movement of the coin in the board allowed are left, right, up and down. The coin movements are defined in the below table:

|  |  |  |
| --- | --- | --- |
| **Movement** | **Current Position** | **Position after move** |
| Left | m, n | m, n-1 if n>0  m, n otherwise |
| Right | m, n | m, n+1 if n<p-1  m, n otherwise |
| Up | m, n | m+1, n if m<p-1  m, n otherwise |
| Down | m, n | m-1, n if m>0  m, n otherwise |

Given a size the board, p, initial cell of the coin in the board, a list of movements and a winning cell in the board, write a code to check if the coin reaches the winning cell from initial cell by these movements. If the winning position is reached then print Win and print Loss otherwise.

Input Format

Dimension of the board, p

Initial cell of the coin m,n separated by a space

Number of moves, num1

Next num1 lines contain the movements l-left, r-right, u-up and d-down

Winning cell of the coin m1, n1 separated by a space

Output Format

Print Win or Loss

**Python Program**

p = int(input())  
temp = input().split(**" "**)  
m = int(temp[0])  
n = int(temp[1])  
moves = []  
board = []  
num1 = int(input())  
for counter in range(num1):  
 moves.append(input().lower())  
temp = input().split(**" "**)  
m1 = int(temp[0])  
n1 = int(temp[1])  
for counter in range(num1):  
 if moves[counter] == **'l'**:  
 if n > 0:  
 n -= 1  
 elif moves[counter] == **'r'**:  
 if n < (p-1):  
 n += 1  
 elif moves[counter] == **'u'**:  
 if m < (p-1):  
 m += 1  
 elif moves[counter] == **'d'**:  
 if m > 0:  
 m -= 1  
 if m == m1 and n == n1:  
 print(**"Win"**)  
 break  
if m != m1 and n != n1:  
 print(**"Loss"**)