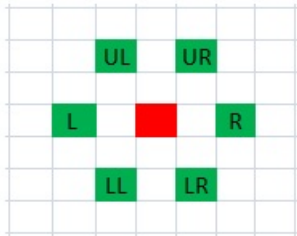


In ordinary chess, the pieces are only of two colors, black and white. In our version of chess, we are including new pieces with unique movements. One of the most powerful pieces in this version is the *red knight*.

The red knight can move to six different positions based on its current position (UpperLeft, UpperRight, Right, LowerRight, LowerLeft, Left) as shown in the figure below.



The board is a grid of size $n \times n$. Each cell is identified with a pair of coordinates (i, j) , where i is the row number and j is the column number, both zero-indexed. Thus, $(0, 0)$ is the upper-left corner and $(n - 1, n - 1)$ is the bottom-right corner.

Complete the function `printShortestPath`, which takes as input the grid size n , and the coordinates of the starting and ending position (i_{start}, j_{start}) and (i_{end}, j_{end}) respectively, as input. The function does not return anything.

Given the coordinates of the starting position of the red knight and the coordinates of the destination, print the minimum number of moves that the red knight has to make in order to reach the destination and after that, print the order of the moves that must be followed to reach the destination in the shortest way. If the destination cannot be reached, print only the word "Impossible".

Note: There may be multiple shortest paths leading to the destination. Hence, assume that the red knight considers its possible neighbor locations in the following order of priority: *UL, UR, R, LR, LL, L*. In other words, if there are multiple possible options, the red knight prioritizes the first move in this list, as long as the shortest path is still achievable. Check sample input **2** for an illustration.

Input Format

The first line of input contains a single integer n . The second line contains four space-separated integers $i_{start}, j_{start}, i_{end}, j_{end}$. (i_{start}, j_{start}) denotes the coordinates of the starting position and (i_{end}, j_{end}) denotes the coordinates of the final position.

Constraints

- $5 \leq n \leq 200$
- $0 \leq i_{start}, j_{start}, i_{end}, j_{end} < n$
- the starting and the ending positions are different

Output Format

If the destination can be reached, print two lines. In the first line, print a single integer denoting the minimum number of moves that the red knight has to make in order to reach the destination. In the second line, print the space-separated sequence of moves.

If the destination cannot be reached, print a single line containing only the word `Impossible`.

Sample Input 0

```
7
6 6 0 1
```

Sample Output 0

```
4
UL UL UL L
```

Explanation 0

	0	1	2	3	4	5	6
0		4		3			
1							
2					2		
3							
4						1	
5							
6							0

Sample Input 1

6
5 1 0 5

Sample Output 1

Impossible

Explanation 1

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Sample Input 2

7
0 3 4 3

Sample Output 2

2
LR LL

Explanation 2

	0	1	2	3	4	5	6
0				0			
1							
2			1		1		
3							
4				2			
5							
6							