ICPC PACNW 2016 L - Windy Path

Time limit: 1.4s **Memory limit:** 256M

There are n obstacles placed in a field. Your task is to design a course that visits each obstacle exactly once, in any order, following a straight line between consecutive obstacles, without ever crossing itself.

The catch? The sequence of turn directions (left or right) has already been decided, in a string of length $n-2$. If the ith character of the turn sequence is `L`, then the locations of the ith, $(i+1)$th, and $(i+2)$th obstacles, in that order must form a counterclockwise angle. If it is `R`, they must form a clockwise angle.

Input

The first line of input contains a single integer n $(3 \leq n \leq 50)$.

Each of the next n lines contains two space-separated integers x_i and y_i $(1 \leq x_i, y_i \leq 1000)$, giving the coordinates of obstacle i.

The next and final line contains a single string with exactly $n-2$ characters consisting of only `L` and `R`, representing the sequence of turn directions.

It is guaranteed that no three obstacles will be collinear.

Output

If no solution is possible, print, on a single line, the integer `-1`. Otherwise, print, on a single line, any permutation of the obstacles that satisfies the requirements. The permutation should be given as n distinct space-separated integers p_i with $1 \leq p_i \leq n$, and this ordering of the points should satisfy the turn directions indicated by the turn sequence.

If there are multiple possible solutions, print any of them.

Sample Input

```
4
2 2
2 1
1 2
1 1
LR
```

Sample Output

```
1 3 2 4
```