

# Giving Directions in Harbin

Input file:            **standard input**  
Output file:         **standard output**  
Time limit:          1 second  
Memory limit:       1024 megabytes

In some regions, people are more accustomed to giving directions using cardinal directions, such as: go south to the second intersection, then head east to the first intersection. However, due to the complex road network planning in Harbin, many streets do not align perfectly with cardinal directions. Thus, if you provide directions using absolute directions to someone who has lived in Harbin for a long time, they may struggle to understand your intended route.

In Harbin, people are more accustomed to using relative directions to give guidance. For the same location, a Harbin resident might first instruct you to face south, and then say: walk straight along the road to the second intersection, then turn left, and then straight to the first intersection.

To address this difference, you decide to write a program that converts the direction-giving style using cardinal directions into the style preferred by Harbin residents. Of course, using a real map of Harbin would be too complicated, so in this problem, you can assume the map is an infinitely large grid.

## Input

The first line contains an integer  $T$  ( $1 \leq T \leq 10^4$ ), indicating the number of test cases.

For each test case, the first line contains an integer  $n$  ( $1 \leq n \leq 10$ ), indicating the number of direction instructions.

The next  $n$  lines each describe an instruction in absolute position, consisting of a character  $d$  ( $d \in \{N, S, W, E\}$ ) and an integer  $x$  ( $1 \leq x \leq 10$ ), indicating “go to the  $x$ -th intersection in the  $d$  direction.” Here, N represents north, S represents south, W represents west, and E represents east.

It is guaranteed that two consecutive instructions will not have the same direction or opposite directions (north and south are opposite, as are west and east).

## Output

For each test case, the first line outputs an integer  $m$  ( $1 \leq m \leq 20$ ) and a character  $f$  ( $f \in \{N, S, W, E\}$ ), representing the number of instructions in Harbin style and the initial facing direction, with the same meanings for directions as in the input.

Next, output  $m$  lines. Each line starts with a character  $g \in \{Z, L, R\}$ , where Z means to go straight, L means to turn left, and R means to turn right. If the character is Z, the line must also include an integer  $y$  ( $1 \leq y \leq 100$ ), representing going straight to the  $y$ -th intersection. The first output instruction must start with Z. Consecutive instructions cannot have the same character  $g$ , and L and R instructions cannot be adjacent.

In this problem, you do not need to minimize  $m$ . If there are multiple ways to reach the same destination, any valid solution is acceptable.

## Example

standard input	standard output
1	3 S
2	Z 2
S 2	L
E 1	Z 1