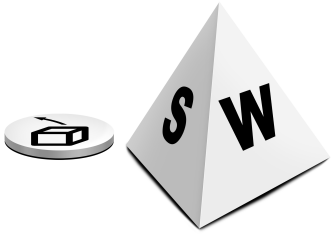
Problem I. Game with coins

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds (3 for java)
Memory limit:	256 MiB
Queries limit:	60 000



This problem is interactive.

Taja can easily win this game, but all of her friends couldn't. Now she offers you to play this game.

Game equipment consists of field $n \times n$ ($5 \le n \le 40$), the piece, two coins (`COIN1`, `COIN2`), two dices with cardinal directions (`DICE1`, `DICE2`), and a lot of one cell size blocks.

Name	Number of faces	Faces
`COIN1`. Move coin	2	SLIDE, RAM
`COIN2`. Modification coin	2	PLACE, REMOVE
`DICE1`. First dice with directions	4	`N` (North), `S` (South), `W` (West), `E` (East)
`DICE2`. Second dice with directions	8	`N` (North), `S` (South), `W` (West), `E` (East), `NW` (Northwest), `NE` (Northeast), `SW` (Southwest), `SE` (Southeast)

Before the beginning of the game the piece and some blocks are placed on the field. Then the player makes moves such kind. At first, the player chooses a coin and tosses it for determine an action. Then he chooses one of the dices and rolls it for determine direction *dir*. After that one of the four action happens:

Coin string	Action
`SLIDE`	Move the piece along empty cells in the direction of *dir*, until the piece collides a block or maze border
`RAM`	Move the piece along empty cells in the direction of *dir*, until it collides a first block. Then move the piece and the block in the same direction, until moving block collides another block or field border
`PLACE`	If adjacent cell in the direction *dir* next to the piece is empty and is in the field, place a block on this cell
`REMOVE`	If adjacent cell in the direction *dir* next to the piece contains a block and is in the field, remove the block from this cell

The goal is to put the piece on the finish cell.

One query for this problem is one coin toss and one dice roll.

Interaction protocol

At first, interactor gives the size of the maze, the maze and coordinate of the finish cell. Then the following 4-step actions are happens:

1. Jury's program displays coordinates of the piece or inform that the piece has reaches the finish cell.

2. Your program displays the coin name.
3. Jury's program displays the name of the action on the coin.
4. Your program displays the dice name.
5. Jury's program displays the name of the direction on the dice and additional information for action "RAM".

Output

The standard output should consist of pair of lines: coin name and dice name. Coin name is the string "COIN1" or "COIN2". Dice name is the string "DICE1" or "DICE2".

Don't forget to flush the standard output after printing each line.

Input

First line of the standard input contains one integer number n — the size of the maze. Next n lines contains n characters "." (ASCII 46) or «#» (ASCII 35), denoting empty cell and cell with block respectively.

Next line contains two integer numbers r_f and c_f — row number and column number of the finish cell. The upper-left corner is $(1, 1)$, the bottom-right is (n, n). Finish cell and initial cell are empty.

Next groups describes each move:

- The first line of an group contains two integer numbers r, c — row number and column number of the piece, or $(-1, -1)$, if the peace reaches the finish cell.
- Next line contains action name shown on the coin.
- Next line contains direction name shown on the dice.
- If the current action is "RAM", next two strings contains two integer numbers r_1, c_1 and r_2, c_2, denotes then the piece collides a block with coordinates (r_1, c_1) and moves that block to the cell (r_2, c_2).

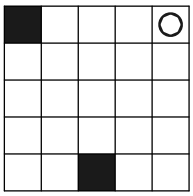
The north direction corresponds to decreasing row number. The south to increasing row number. The west direction to decreasing column number. The east direction to increasing column number.

Each side of any coin or dice is shown with the same probability.

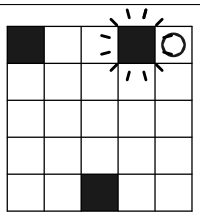
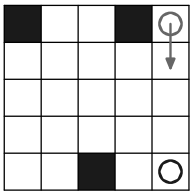
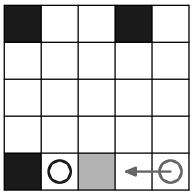
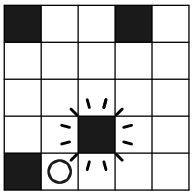
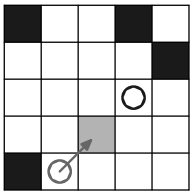
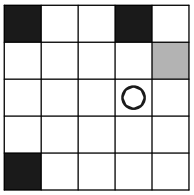
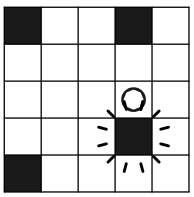
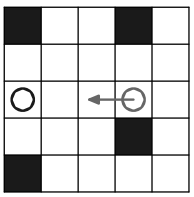
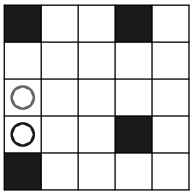
Examples

standard output	standard input
	5 #.... ....#.. 4 3 1 5
COIN2 DICE1	PLACE W 1 5
COIN1 DICE1	RAM S 6 5 6 5 5 5
COIN1 DICE1	RAM W 5 3 5 1 5 2
COIN2 DICE2	PLACE NE 5 2
COIN1 DICE2	RAM NE 4 3 2 5 3 4
COIN2 DICE2	REMOVE NE 3 4
COIN2 DICE1	PLACE S 3 4
COIN1 DICE1	SLIDE W 3 1
COIN1 DICE1	RAM S 5 1 5 1 4 1
COIN1 DICE1	SLIDE E -1 -1

The initial state of the maze for the sample is



And the representation of the path is shown below:

COIN2 → PLACE DICE1 → W		COIN1 → RAM DICE1 → S	
COIN1 → RAM DICE1 → W		COIN2 → PLACE DICE2 → NE	
COIN1 → RAM DICE2 → NE		COIN2 → REMOVE DICE2 → NE	
COIN2 → PLACE DICE1 → S		COIN1 → SLIDE DICE1 → W	
COIN1 → RAM DICE1 → S		COIN1 → SLIDE DICE1 → E