Problem E. Maze with a Hint

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

Thor had boasted to dwarves that he can go through any maze without a single drop of sorcery, using only a small torch. The dwarves decided to give Thor a trial. They are fast and skilled builders, and their new maze will be large and tricky. If an adventurer walks inside for too long, the torch will go out, and the dwarves will laugh at the Ace. After taking a look on the new construction, Thor decided that he has to win, no matter the price, and asked Loki for help.

Loki is a resourceful trickster, and he will be able to obtain the map of the maze as soon as it is finished. But he won't be able to just give the map to Thor: the dwarves will surely see through such deception. Thor can only get a short hint from Loki...

Help Thor and Loki to prepare for passing the hint, so that Thor would then be able to pass the maze before the torch goes out.

Maze structure

The maze which is being built by dwarves can be drawn as a square board consisting of $n \times n$ squares. Between each two squares adjacent horizontally or vertically, there is either a passage or a wall. Additionally, the whole maze is surrounded by a wall. The entrance is in the bottom left square, and the exit is in the top right square.

The dwarves build walls as follows. They consider all possible positions for the walls inside the maze in random order, each position exactly once. In each such position, they erect a wall if after that, it is still possible to move from every square of the maze to every other square.

In text form, the maze is given by $2n + 1$ lines, each containing $2n + 1$ characters. The even rows and columns, if counted from one, correspond to squares, and the odd ones to the walls between them. The "`.`" character corresponds to a square or a passage, and the "`#`" character to a wall or a joint between walls. In particular, a character in an even row and an even column is always a dot (it's a square), and a character in an odd row and an odd column is always a hash (it's a joint between walls).

A map of a 5×5 maze using the described notation can be seen in the example below. Additionally, to make local testing easier, you can download all mazes from tests with odd numbers. They can be found under "Samples ZIP" in the testing system interface.

Interaction Protocol

In this problem, your solution will be run twice on each test.

During the first run, the solution obtains a map and writes the hint as Loki. The first line contains the word "`view`". The second line contains an integer n, the size of the maze ($5 \le n \le 200$). Each of the next $2n + 1$ lines contains $2n + 1$ characters. Together these lines constitute the map of the maze.

The solution has to print one line: the hint that Loki will send to Thor. This hint has to be composed from digits `0` and `1` and have length from 0 to 1000 characters.

During the second run, the solution obtains the hint and walks through the maze as Thor. This run is interactive. The first line contains the word "`walk`". The second line contains the hint given by Loki to Thor: the one printed by the solution during the first run. The third line contains the integer n, the size of the maze, same as during the first run.

After that, the solution will get a piece of the map seen by Thor with the help of his torch, and in response, it should print the direction of his next step. Each piece of the map is given as three lines containing three characters each: the square of the maze where Thor is, along with its surroundings.

If the solution is sure that Thor passed the maze and stands at the exit, the solution should simply terminate gracefully. Otherwise, it has to print a line containing the direction of Thor's next step: "`N`" for a step to the North (up on the map), "`W`" for a step to the West (left on the map), "`S`" for a step to the South (down on the map), or "`E`" for a step to the East (right on the map). After that, the solution should flush the output buffer: this can be done by calling, for example, `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, or `sys.stdout.flush ()` in Python.

If the passage is clear, Thor moves to an adjacent square in the requested direction and is given the next piece of the map seen by him at the moment. If there is a wall in the given direction, the process is terminated with "`Wrong Answer`" outcome.

The solution passes the maze if it terminated gracefully when Thor stood at the exit, and made at most 6000 steps before that.

Example

On each test, the input during the second run depends on the solution's output during the first run. In the example, We will consider solution which forms the hint by simply printing the required sequence of steps: `0` for a step to the East and `1` for a step to the North. Sure enough, this solution does not always work.

Two runs of this solution on the first test are shown below. In the second run, blank lines are added to show the sequence of events.

standard input	standard output
view 5 ########### #.#.......# #.###.##### #.#.....#.# #.#.###.#.# #...#.....# ###.####### #.........# ###.####### #.........# ###########	10001011
walk 10001011 5 ### #.. ### #.# ... ### #.# ... #.# #.# ..# #.# ### #.. #.# #.# ... ### ### ... #.# ### ... ### ### ..# ###	 E N N N E N E E