Amazing Robots

TASK

You are the proud owner of two robots that are located in separate rectangular mazes. Square (1, 1) in a maze is the square in the upper-left corner, which is the north-west corner. Maze i (i = 1, 2) has a set of G_i ($0 \le G_i \le 10$) guards trying to capture the robots by patrolling back and forth on a straight patrol path. Your goal is to determine a sequence of commands such that the robots exit the mazes without either robot being captured by a guard.

At the beginning of each minute, you broadcast the same command to both robots. Each command is a direction (north, south, east, or west). A robot moves one square in the direction of the command, unless the robot would collide with a wall, in which case the robot does not move for that minute. A robot exits the maze by walking out of it. A robot ignores commands after it has exited its maze.

Guards move one square at the beginning of each minute, at the same time as the robots. Each guard begins at a given square facing a given direction and moves forward one square per minute until the guard has moved one fewer square than the number of squares in his patrol path. The guard then turns around instantaneously and walks in the opposite direction back to his starting square, where he turns around again and repeats his patrol path until each robot has exited its maze.

A guard's patrol will not require the guard to walk through walls or exit the maze. Although guard patrols may overlap, no two guards will ever collide: they will never occupy the same square at the end of a minute, and they will not exchange squares with each other during a minute. A guard in a maze will not start in the same square as the robot in that maze.

A guard captures a robot if the guard occupies the same square at the end of a minute as the robot, or if the guard and the robot exchange squares with each other during a minute.

Given two mazes (each no larger than 20x20) along with the initial square of each robot and the patrol paths of the guards in each maze, determine a sequence of commands for which the robots exit without being caught by the guards. Minimize the time it takes for the later robot to exit its maze. If the robots exit at different times, the time at which the earlier robot exited does not matter.

Input: `robots.in`

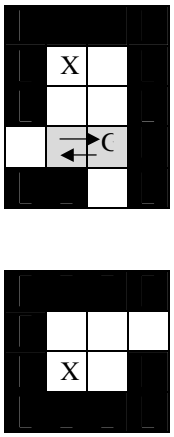
The first set of lines describes the first maze and its occupants. Subsequently, the second set of lines describes the second maze and its occupants.

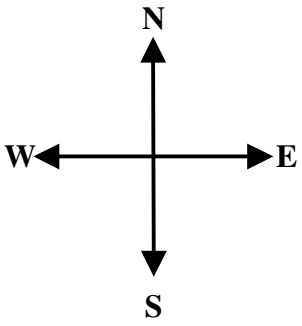
- The first line of input contains two space-separated integers, R_1 and C_1, the number of rows and columns in maze 1.
- The next R_1 lines each contain C_1 characters specifying the maze layout. The robot's starting square is specified by an '`X`'. A '`.`' represents an open space and '#' represents a wall. Each maze will contain exactly one robot.
- Following the maze layout is a single line with a single integer G_1, the number of guards in the first maze ($0 \le G_1 \le 10$).
- Each of the next G_1 lines describes a guard's patrol as three integers and a character, separated by single spaces. The first two integers specify the row and column of the starting square of the guard. The third integer specifies the number of squares (2…4) in the guard's patrol path. The character specifies the initial direction the guard is facing: '`N`', '`S`', '`E`', or '`W`' (north, south, east, and west, respectively).

The description of the second maze follows the description of the first, in the same format but with potentially different values.

Example input:

```
5 4
####
#X.#
#..#
...#
##.#
1
4 3 2 W
4 4
####
#...
#X.#
####
0
```





Output: `robots.out`

The first line of the output should contain a single integer K ($K \le 10000$), the number of commands for both robots to exit the maze without being captured. If such a sequence of commands exists, the shortest sequence will have no more than 10000 commands. The next K lines are the sequence of commands, each containing a single character from the set {'`N`', '`S`', '`E`', '`W`'}. If no such sequence exists, output a single line containing "`-1`".

Both robots should exit their mazes by the end of the commands. The last command should cause at least one of the robots to exit its maze. If multiple sequences of commands cause the robots to exit in the minimum time, any will be accepted.

Example output:

```
8
E
N
E
S
S
S
E
S
```

CONSTRAINTS

Running time	2 seconds of CPU
Memory	64 MB

SCORING

No partial credit will be given on test cases for which no sequence of commands exists. Partial credit for other test cases will be given as described below.

Correctness: 20% of points

The output file for a test case is considered correct if it is correctly formatted, contains no more than 10000 commands, and the sequence of commands causes the robots to exit the mazes, with the last command causing at least one robot to exit its maze.

Minimality: 80% of points

The output file for a test case is considered minimal if it is correct and there is no shorter sequence of commands that is correct. A program whose sequence of commands is not minimal will receive no points for minimality.