Rules for Bender Episode 1 from CodinGame

- https://www.codingame.com/ide/puzzle/bender-episode-1

Goal

Bender is a depressed robot who heals his depression by partying and drinking alcohol. To save him from a life of debauchery, his creators have reprogrammed the control system with a more rudimentary intelligence. Unfortunately, he has lost his sense of humor and his former friends have now rejected him.

Bender is now all alone and is wandering through the streets of Futurama with the intention of ending it all in a suicide booth.

To intercept him and save him from almost certain death, the authorities have given you a mission: write a program that will make it possible to foresee the path that Bender follows. To do so, you are given the logic for the new intelligence with which Bender has been programmed as well as a map of the city.

Rules

The 9 rules of the new Bender system:

- Bender starts from the place indicated by the @ symbol on the map and heads SOUTH.
- Bender finishes his journey and dies when he reaches the suicide booth marked $.
- Obstacles that Bender may encounter are represented by # or X.
- When Bender encounters an obstacle, he changes direction using the following priorities: SOUTH, EAST, NORTH and WEST. So he first tries to go SOUTH, if he cannot, then he will go EAST, if he still cannot, then he will go NORTH, and finally if he still cannot, then he will go WEST.
- Along the way, Bender may come across path modifiers that will instantaneously make him change direction. The S modifier will make him turn SOUTH from then on, E, to the EAST, N to the NORTH and W to the WEST.
- The circuit inverters (I on map) produce a magnetic field which will reverse the direction priorities that Bender should choose when encountering an obstacle. Priorities will become WEST, NORTH, EAST, SOUTH. If Bender returns to an inverter I, then priorities are reset to their original state (SOUTH, EAST, NORTH, WEST).
- Bender can also find a few beers along his path (B on the map) that will give him strength and put him in “Breaker” mode. Breaker mode allows Bender to destroy and automatically pass through the obstacles represented by the character X (only the obstacles X). When an obstacle is destroyed, it remains so permanently and Bender maintains his course of direction. If Bender is in Breaker mode and passes over a beer again, then he immediately goes out of Breaker mode. The beers remain in place after Bender has passed.
- 2 teleporters T may be present in the city. If Bender passes over a teleporter, then he is automatically teleported to the position of the other teleporter and he retains his direction and Breaker mode properties.
- Finally, the space characters are blank areas on the map (no special behavior other than those specified above).

Your program must display the sequence of moves taken by Bender according to the map provided as input.

The map is divided into lines (L) and columns (C). The contours of the map are always unbreakable # obstacles. The map always has a starting point @ and a suicide booth $.

If Bender cannot reach the suicide booth because he is indefinitely looping, then your program must only display LOOP.

Example

Let the map below :

```
######
#@E $#
# N #
#X #
######
```

In this example, Bender will follow this sequence of moves:

- SOUTH (initial direction)
- EAST (because of the obstacle X)
- NORTH (change of direction caused by N)
- EAST (change of direction caused by E)
- EAST (current direction, until end point $)

Game Input

Input

- Line 1: the number of lines L and columns C on the map, separated by a space.
- The following L lines: a line of the length C representing a line on the map. A line can contain the characters #, X, @, $, S, E, N, W, B, I, T and space character.

Output

- If Bender can reach $, then display the sequence of moves he has taken. One move per line: SOUTH for the South, EAST for the East, NORTH for the North and WEST for the west.
- If Bender cannot reach $, then only display LOOP.

Constraints

- $4 \leq C \leq 100$
- $4 \leq L \leq 100$