Tile Traveller
Problem ID: a08p01tiletraveller

The assignment is to develop a computer game in which a player is located in a grid of tiles, one of which is occupied by the player. At each iteration, the program displays the directions for which there are adjacent tiles that the player can travel to.

The program only displays text, so you do not actually draw the tile grid, but the program should behave as if the player is in a 3×3 grid, as seen in the following image:

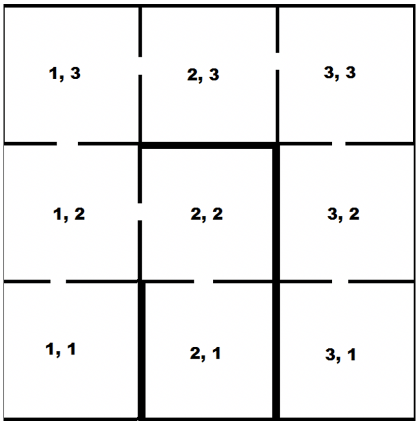

Figure 1: `Maze`

As indicated in the image, there are walls between some adjacent tiles, blocking the player from travelling directly between them. In other cases, there's an open path from one tile to the next. You may think of these open paths as doorways, or just the absence of walls, it does not matter, the effect is the same.

The player starts in tile `(1, 1)`. At the beginning, and after each move selected by the player, the program should print the player's travel options. If there is no wall in a given direction, write that direction as a possible travel direction.

At each iteration, the player enters the first letter of the direction in which they wish to travel, after which the player should be located in another tile and the options for the new tile are then printed out.

The order in which the directions should be listed, is *North, East, South, West*.

The player enters:

- n/N for North (up)
- e/E for East (right)
- s/S for South (down)
- w/W for West (left)

If the player enters an invalid direction, the program prints "`Not a valid direction!`" and prompts the player to enter the direction again.

For example, in tile `(1,1)` it is only possible to move north. In tile `(1,2)` the possible moves are north, east, and south. In tile `(3,3)` the possible moves are south and west.

Tile `(3,1)` is the victory location When the player enters this tile, the program should notify them of their victory and quit.

Input

Input consists of one line for each move made by the player, in order, where each move consists of any of the characters `n`, `e`, `s`, `w`, `N`, `E`, `S`, or `W`. These lines repeat until the player has won the game. It is guaranteed that the player will win the game in less than 200 moves.

Output

First, the program should print "`You can travel: {Valid directions}.`", without quotations, where valid directions can be `(N)orth`, `(E)ast`, `(S)outh`, and `(W)est`. The directions should be separated by spaces and the word "`or`". The following repeats until the user wins the game, at which point the program exits after outputting a victory message:

- The program prompts the user with "`Direction: `", without quotations.
- If the player inputs an invalid direction, the program should output "`Not a valid direction!`", without quotations.
- If the player inputs a valid direction, the program should output the new valid directions as stated above, "`You can travel: {Valid directions}.`", without quotations.
- If the player wins, the program should output "`Victory!`", without quotations.

Sample Input 1

```
s
n
N
w
E
e
s
S
```

Sample Output 1

```
You can travel: (N)orth.
Direction: Not a valid direction!
You can travel: (N)orth.
Direction: You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (E)ast or (S)outh.
Direction: Not a valid direction!
You can travel: (E)ast or (S)outh.
Direction: You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (N)orth or (S)outh.
Direction: Victory!
```

Sample Input 2

```
n
w
E
s
N
w
N
e
E
w
e
s
n
s
S
```

Sample Output 2

```
You can travel: (N)orth.
Direction: You can travel: (N)orth or (E)ast or (S)outh.
Direction: Not a valid direction!
You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (N)orth.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (E)ast or (S)outh.
Direction: You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (N)orth or (S)outh.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (N)orth or (S)outh.
Direction: Victory!
```

Sample Input 3

```
s
n
s
e
n
n
e
e
w
e
s
s
```

Sample Output 3

```
You can travel: (N)orth.
Direction: Not a valid direction!
You can travel: (N)orth.
Direction: You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (N)orth.
Direction: Not a valid direction!
You can travel: (N)orth.
Direction: You can travel: (N)orth or (E)ast or (S)outh.
Direction: You can travel: (E)ast or (S)outh.
Direction: You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (E)ast or (W)est.
Direction: You can travel: (S)outh or (W)est.
Direction: You can travel: (N)orth or (S)outh.
Direction: Victory!
```