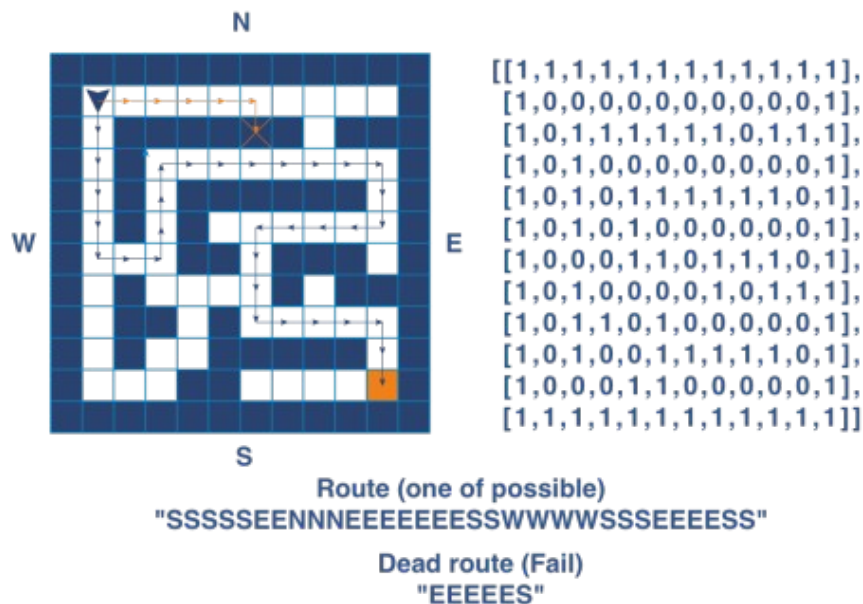


## open labyrinth

The labyrinth has no walls, but pits surround the path on each side. If a player falls into a pit, they lose. The labyrinth is presented as a matrix (a list of lists): 1 is a pit and 0 is part of the path. The labyrinth's size is 12 x 12 and the outer cells are also pits. Players start at cell (1,1). The exit is at cell (10,10). You need to find a route through the labyrinth. Players can move in only four directions--South (down [1,0]), North (up [-1,0]), East (right [0,1]), West (left [0, -1]). The route is described as a string consisting of different characters: "S"=South, "N"=North, "E"=East, and "W"=West.



Input: A labyrinth's map. A list of lists with 1 and 0.

Output: A route. A string that contain "W", "E", "N" and "S".

Example:

```
checkio([
  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
  [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1],
  [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1],
  [1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1],
  [1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1],
  [1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1],
  [1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1],
  [1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1],
  [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1],
  [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])
```

How it is used: This is a classical problem for searching a path in graphs. It can be used for navigation software and the gaming industry.