Sprint 11: Combine Arrays and Functions to solve problems

Problem Statement: 11.1

To win a competition, a contestant has to enter the maze and reach the exit without running into any obstacle. Positions in the maze are either open or blocked by an obstacle. To reach the exit and to locate paths containing only open cells, the participant has to find positions without obstacles. Retracing one or more cells is not considered a new path. At any given moment, the runner can move only one-step in any one of the four directions; s/he can move either north, south, east or west.

```
Go North (x,y) -> (x-1,y)
Go South (x,y) -> (x+1,y)
Go East (x,y) -> (x,y+1)
Go West (x,y) -> (x,y-1)
```

Hints :

- From the starting position 'S', first try to go one space towards the North and then recursively try to solve from there.
- If the attempt is not successful, then take a step towards the South and repeat it recursively.
- If moving towards the South does not work, then try to solve the maze by moving a step towards the West.
- If North, South, and West have not been successful recursively, solve from the East.

Write a Java program to find a unique path in the maze from the start to the end point. Use the hints mentioned in the problem statement above to solve the maze. The path should be denoted using the ‘*’ symbol.

Sample Output:

```
The solved maze skeleton
-----------------

‡*‡‡‡‡‡‡‡‡‡‡‡
‡*‡ ‡ ‡     ‡
‡*‡   ‡ ‡‡‡ ‡
‡*  ‡‡‡*****‡
‡*‡    *‡‡‡*‡
‡*‡ ‡‡‡*‡  *‡
‡*‡ ‡  *‡‡‡*‡
‡*‡ ‡‡‡*‡ ‡*‡
‡*******  ‡E‡
‡‡‡‡‡‡‡‡‡‡‡‡‡
```

Practice Challenge – 11.1 – Boilerplate URL

https://myrepos.stackroute.niit.com/core_java_boilerplates/sprint11_pc11.1.git