Code Test

Instructions: Please complete the exercise below, paying particular attention to the use of tests to provide functional assurance, document behaviour and guide your solution design. Feel free to use any language listed on the job description you are applying for. Please email either a zip archive containing your finished code or else a link to your solution VCS repo in reply to the address from which you received this test.
Time Allocated: 3hrs

Maze Solver
Consider a 5x5 grid maze, with a positive integer co-ordinate system and the point [0,0] located in the bottom left corner. Points are specified in [x,y] format. The grid has an entrance at position [0,1] and an exit at position [4,3] as shown below:

```
    *****
    *    -> Out
    *   *
In->    *
    *****
```

All mazes have a core definition as above. In addition, internal dividers can be described within the grid using the format [x,y]:[x',y'] to specify start and end points. Multiple dividers can be specified in semi-colon delimited format. For example:

1. The maze [2,0]:[2,2] looks like

```
    *****
    *
    * * *
      * *
    *****
```

2. The maze [0,2]:[1,2];[3,2]:[4,2] looks like

```
    *****
    *
    ** **
        *
    *****
```

3. And the maze [2,2]:[2,3];[2,2]:[3,2] looks like

```
    *****
    * *
    * ***
        *
    *****
```

In a language of your choice from the job description, write a tested program capable of identifying the path from entrance to exit for any maze of the type described above. If the maze can be solved, the program should output a valid path from entrance to exit (any valid path is equally good) else it should output the string 'blocked'. So for the examples above, the outputs should be:

1. [0,1]
   [1,1]
   [1,2]
   [1,3]
   [2,3]
   [3,3]
   [4,3]

2. [0,1]
   [1,1]
   [2,1]
   [2,2]
   [2,3]
   [3,3]
   [4,3]

3. blocked