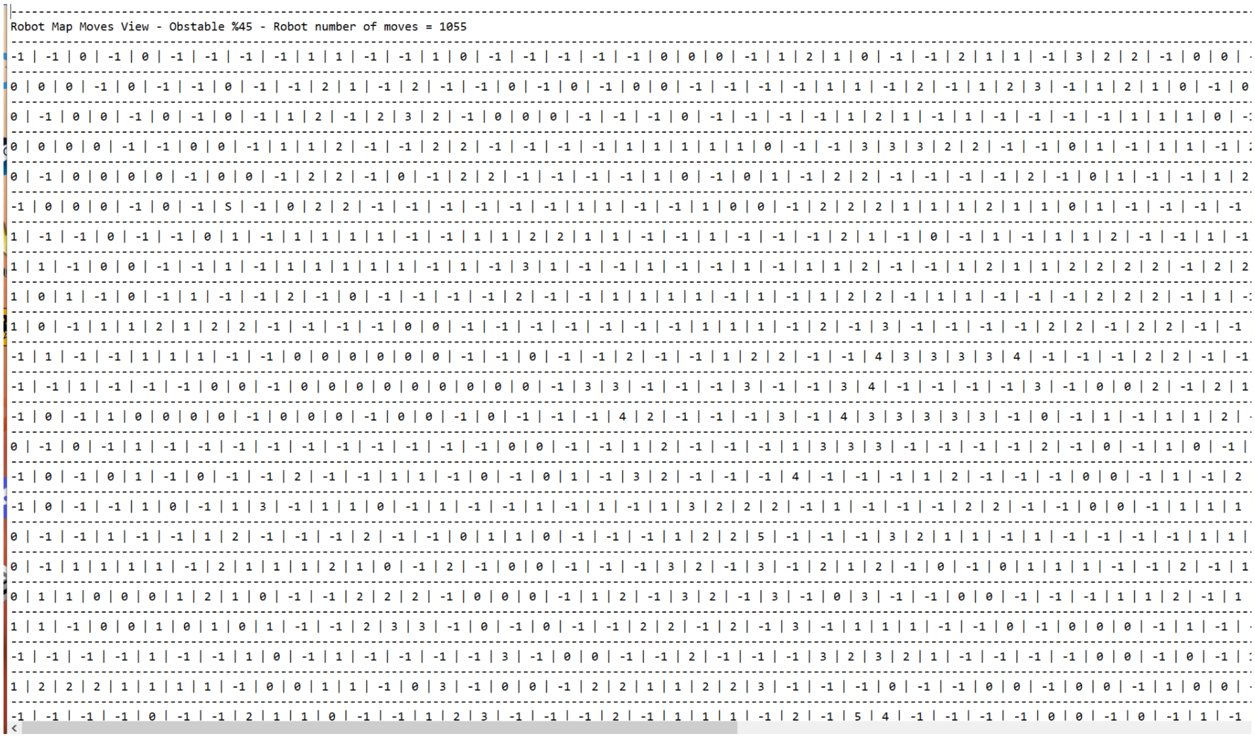


```
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------
Robot Map Moves View - Obstable %45 - Robot number of moves = 1055
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | -1 | 0 | -1 | 0 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 | 0 | -1 | -1 | -1 | -1 | -1 | 0 | 0 | 0 | -1 | 1 | 2 | 1 | 0 | -1 | -1 | 2 | 1 | 1 | -1 | 3 | 2 | 2 | -1 | 0 | 0 | -
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | 0 | 0 | -1 | 0 | -1 | -1 | 0 | -1 | -1 | 2 | 1 | -1 | 2 | -1 | -1 | 0 | -1 | 0 | -1 | 0 | 0 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | 2 | -1 | 1 | 2 | 3 | -1 | 1 | 2 | 1 | 0 | -1 | 0
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | -1 | 0 | 0 | -1 | 0 | -1 | 0 | -1 | 1 | 2 | -1 | 2 | 3 | 2 | -1 | 0 | 0 | 0 | -1 | -1 | -1 | 0 | -1 | -1 | -1 | -1 | 1 | 2 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 0 | -
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | -1 | 1 | 1 | 2 | -1 | -1 | 2 | 2 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | 1 | 0 | -1 | -1 | 3 | 3 | 3 | 2 | 2 | -1 | -1 | 0 | 1 | -1 | 1 | 1 | -1 | 2
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | -1 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 2 | 2 | -1 | 0 | -1 | 2 | 2 | -1 | -1 | -1 | -1 | 1 | 0 | -1 | 0 | 1 | -1 | 2 | 2 | -1 | -1 | -1 | -1 | 2 | -1 | 0 | 1 | -1 | -1 | 1 | 2
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | 0 | 0 | 0 | -1 | 0 | -1 | S | -1 | 0 | 2 | 2 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 | 0 | 0 | -1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | -1 | -1 | -1 | -1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
1 | -1 | -1 | 0 | -1 | -1 | 0 | 1 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | 2 | 2 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 2 | 1 | -1 | 0 | -1 | 1 | -1 | 1 | 1 | 2 | -1 | -1 | 1 | -1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
1 | 1 | -1 | 0 | 0 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 3 | 1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | 2 | -1 | -1 | 1 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | -1 | 2 | 2
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
1 | 0 | 1 | -1 | 0 | -1 | 1 | -1 | -1 | 2 | -1 | 0 | -1 | -1 | -1 | -1 | 2 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | 1 | -1 | 1 | 2 | 2 | -1 | 1 | 1 | -1 | -1 | -1 | 2 | 2 | 2 | -1 | 1 | -
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
1 | 0 | -1 | 1 | 1 | 2 | 1 | 2 | 2 | -1 | -1 | -1 | -1 | 0 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 2 | -1 | 3 | -1 | -1 | -1 | -1 | 2 | 2 | -1 | 2 | 2 | -1 | -1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | -1 | -1 | 2 | -1 | -1 | 1 | 2 | 2 | -1 | -1 | 4 | 3 | 3 | 3 | 3 | 4 | -1 | -1 | -1 | 2 | 2 | -1 | -1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | -1 | 1 | -1 | -1 | -1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 3 | 3 | -1 | -1 | -1 | 3 | -1 | -1 | 3 | 4 | -1 | -1 | -1 | -1 | 3 | -1 | 0 | 0 | 2 | -1 | 2 | 1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | 0 | -1 | 1 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | -1 | -1 | -1 | 4 | 2 | -1 | -1 | -1 | 3 | -1 | 4 | 3 | 3 | 3 | 3 | 3 | -1 | 0 | -1 | 1 | -1 | 1 | 1 | 2 | -
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | -1 | 0 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 0 | 0 | -1 | -1 | 1 | 2 | -1 | -1 | -1 | 1 | 3 | 3 | 3 | -1 | -1 | -1 | -1 | 2 | -1 | 0 | -1 | 1 | 0 | -1 |
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | 0 | -1 | 0 | 1 | -1 | 0 | -1 | -1 | 2 | -1 | -1 | 1 | 1 | -1 | 0 | -1 | 0 | 1 | -1 | 3 | 2 | -1 | -1 | -1 | 4 | -1 | -1 | -1 | 1 | 2 | -1 | -1 | -1 | 0 | 0 | -1 | 1 | -1 | 2
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | 0 | -1 | -1 | 1 | 0 | -1 | 1 | 3 | -1 | 1 | 1 | 0 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | -1 | 1 | 3 | 2 | 2 | 2 | -1 | 1 | -1 | -1 | -1 | 2 | 2 | -1 | -1 | 0 | 0 | -1 | 1 | 1 | 1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | -1 | -1 | 1 | -1 | -1 | 1 | 2 | -1 | -1 | -1 | 2 | -1 | -1 | 0 | 1 | 1 | 0 | -1 | -1 | -1 | 1 | 2 | 2 | 5 | -1 | -1 | -1 | 3 | 2 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 |
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | -1 | 1 | 1 | 1 | 1 | -1 | 2 | 1 | 1 | 1 | 2 | 1 | 0 | -1 | 2 | -1 | 0 | 0 | -1 | -1 | -1 | 3 | 2 | -1 | 3 | -1 | 2 | 1 | 2 | -1 | 0 | -1 | 0 | 1 | 1 | 1 | -1 | -1 | 2 | -1 | 1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 1 | 0 | -1 | -1 | 2 | 2 | 2 | -1 | 0 | 0 | 0 | -1 | 1 | 2 | -1 | 3 | 2 | -1 | 3 | -1 | 0 | 3 | -1 | -1 | 0 | 0 | -1 | -1 | -1 | 1 | 1 | 2 | -1 | 1
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
1 | 1 | -1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | -1 | -1 | 2 | 3 | 3 | -1 | 0 | -1 | 0 | -1 | -1 | 2 | 2 | -1 | 2 | -1 | 3 | -1 | 1 | 1 | 1 | -1 | -1 | 0 | -1 | 0 | 0 | 0 | -1 | 1 | -1 | -
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | 0 | -1 | 1 | -1 | -1 | -1 | -1 | 3 | -1 | 0 | 0 | -1 | -1 | 2 | -1 | -1 | -1 | 3 | 2 | 3 | 2 | 1 | -1 | -1 | -1 | -1 | 0 | 0 | -1 | 0 | -1 |
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | -1 | 0 | 0 | 1 | 1 | -1 | 0 | 3 | -1 | 0 | 0 | -1 | 2 | 2 | 1 | 1 | 2 | 2 | 3 | -1 | -1 | -1 | 0 | -1 | -1 | 0 | 0 | -1 | 0 | 0 | -1 | 1 | 0 | 0 | -
---------------------------------------------------------------------------------------------------------------------------------------------------------------------
-1 | -1 | -1 | -1 | 0 | -1 | -1 | 2 | 1 | 1 | 0 | -1 | -1 | 1 | 2 | 3 | -1 | -1 | -1 | 2 | -1 | 1 | 1 | 1 | -1 | 2 | -1 | 5 | 4 | -1 | -1 | -1 | -1 | 0 | 0 | -1 | 0 | -1 | 1 | -1
```

CS 657 - Assingment 1

09.19.2022

Andrick Mercado

Overview

In this assignment, I developed a program that can rotate in 8 directions and has three sensors to detect obstacles in front of it as well as to its left and right, all for the purpose of reaching and end destination with the least amount of moves given the information the sensors gather will exploring.

Goals

1. Can go on reverse
2. Can reach goal destination
3. Can detect no solutions
4. Make a working interface

Specifications

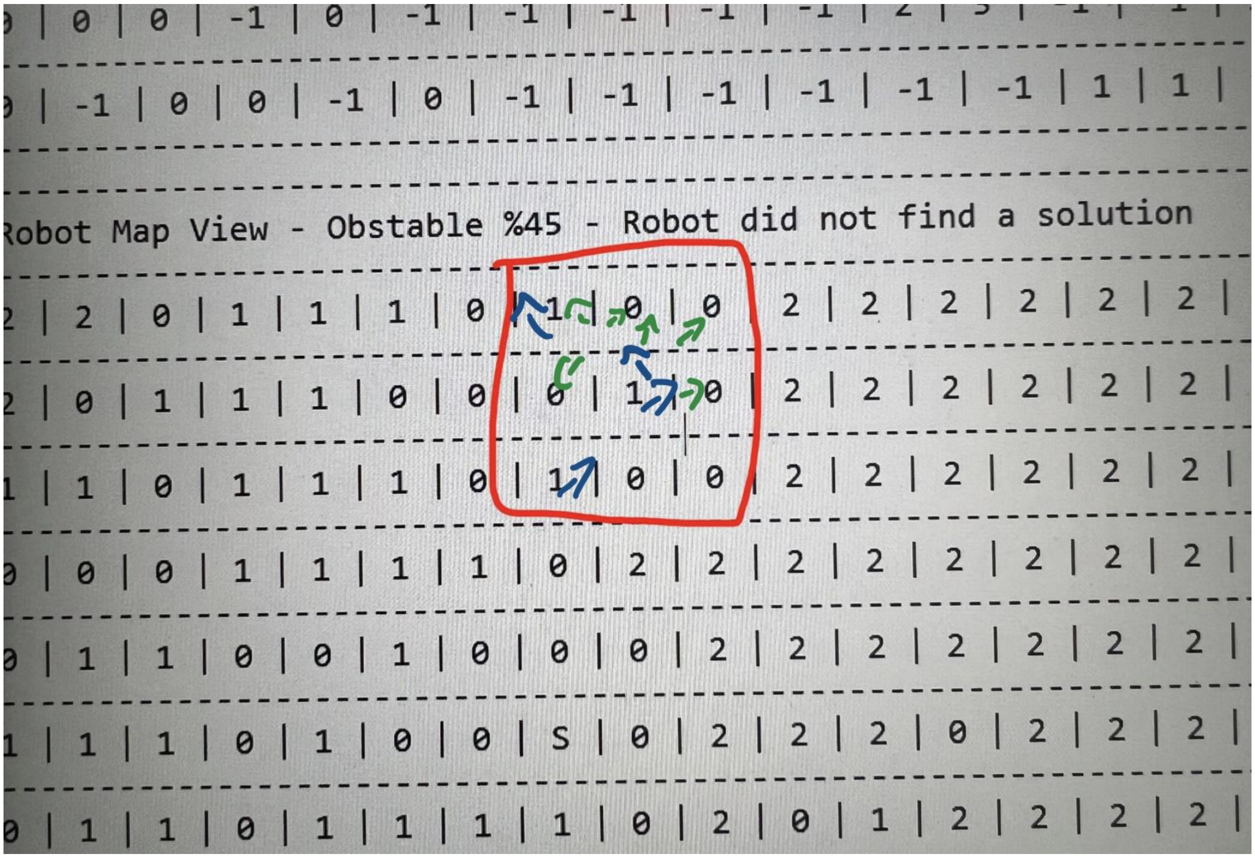
Robot can only go in 8 directions, and move one cell at a time and detect new cells with its sensors and store the data in its memory, rotates out of dead ends or reverse, is coded only using rule-based system made out of if than rules.

Milestones That were not reached

I. Robot was not made to go in reverse

II. Case where it gets stuck

In this case we have an infinite loop where the robot cannot get out of the dead end, this robot was made to always default to left turns and not to rotate in place when there's a legal move, hence why it fails here.



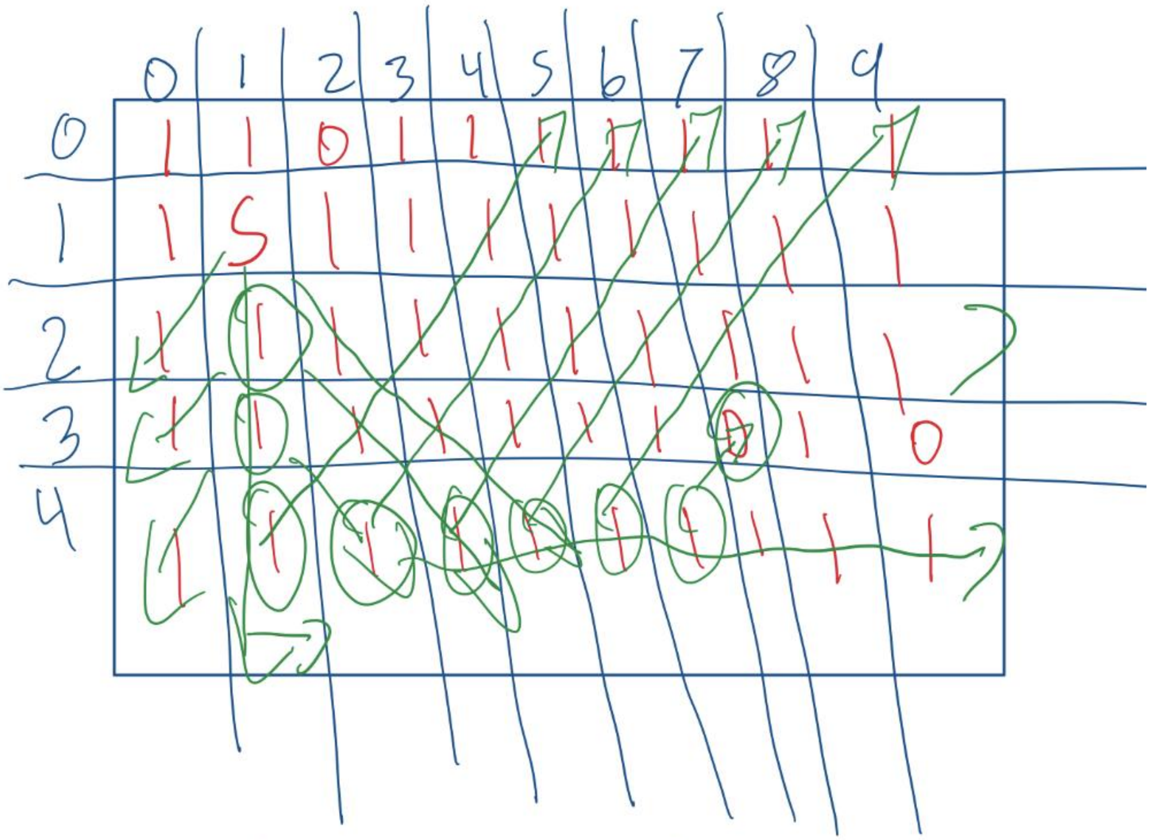
```
0 | 0 | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -1 | 2 | 3 | -1 | 1 |
-------------------------------------------------------------
0 | -1 | 0 | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 |
-------------------------------------------------------------

-------------------------------------------------------------
Robot Map View - Obstable %45 - Robot did not find a solution
-------------------------------------------------------------
2 | 2 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 |
-------------------------------------------------------------
2 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 2 | 2 | 2 | 2 | 2 |
-------------------------------------------------------------
1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 |
-------------------------------------------------------------
0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
-------------------------------------------------------------
0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
-------------------------------------------------------------
1 | 1 | 1 | 0 | 1 | 0 | 0 | S | 0 | 2 | 2 | 2 | 0 | 2 | 2 | 2 |
-------------------------------------------------------------
0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 2 | 2 | 2 | 2 |
```

III. Here are other pictures of my logic as well as test results

IV. Hopefully, this helps understand how I arrived to my solution if any questions feel free to ask

- straight always
- left turns first

Start ↓

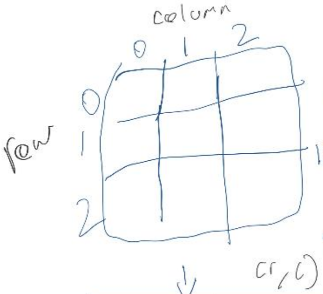


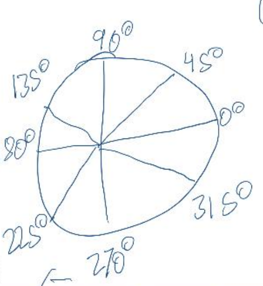
	0	1	2	3	4	5	6	7	8	9
0	1	1	0	1	1	1	1	1	1	1
1	1	S	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	0	1	0
4	1	1	1	1	1	1	1	1	1	1

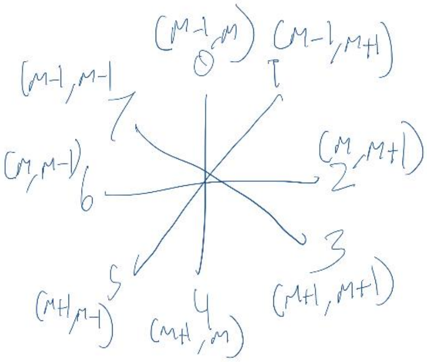
10 moves

Initial check if at a border

- Check borders if in one







Make fu
Call th
to trave

↓ (1,1) at 270°
225° 270° 315°
(2,0) (2,1) (2,2)
(M+1, M-1) (M+1, M) (M+1, M+1)

← (1,1) at 180°
135° 180° 225°
(0,0) (1,0) (2,0)
(M+1, M-1) (M, M-1) (M+1, M-1)

↑ (1,1) at 90°
45° 90° 135°
(0,2) (0,1) (0,0)
(M-1, M+1) (M-1, M) (M-1, M-1)

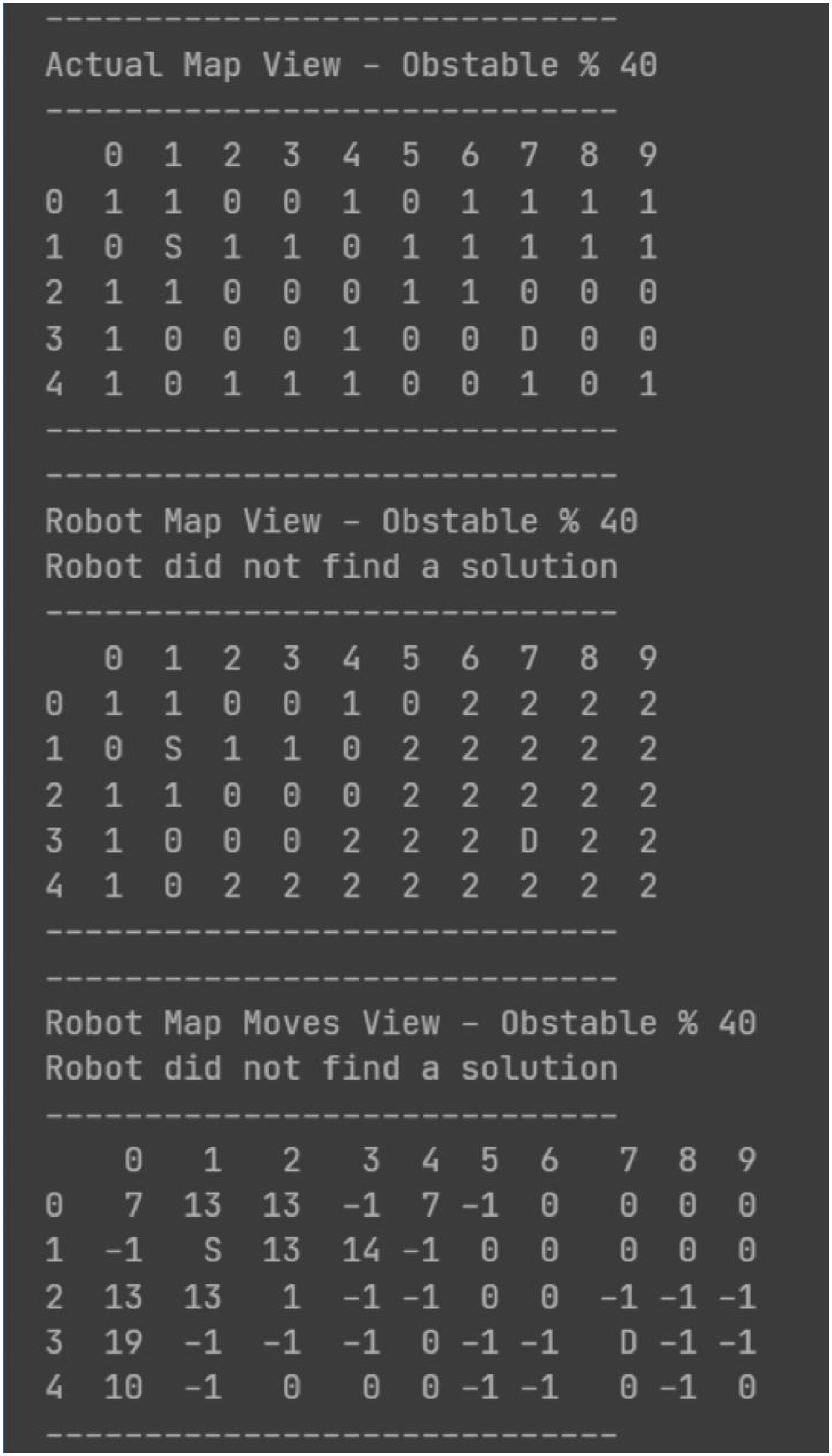
→ (1,1) at 0°
315° 0° 45°
(2,2) (1,2) (0,2)
(M+1, M+1) (M, M+1) (M-1, M+1)

↙ (1,1) at 225°
180° 225° 270°
(1,0) (2,0) (2,1)
(M, M-1) (M+1, M-1) (M+1, M)

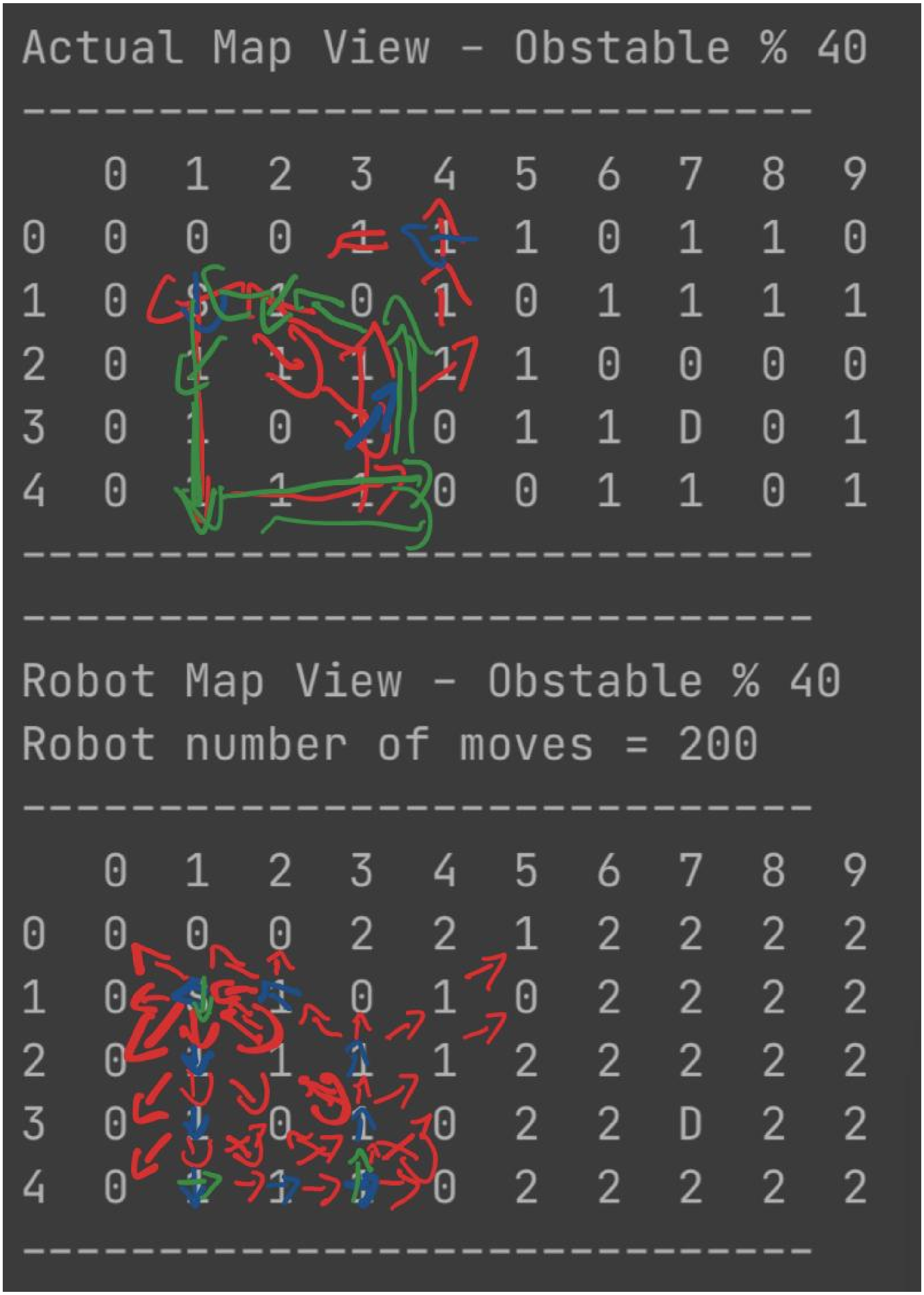
↖ (1,1) at 135°
90° 135° 180°
(0,1) (0,0) (1,0)
(M-1, M) (M-1, M-1) (M, M-1)

↗ (1,1) at 45°
0° 45° 90°
(1,2) (0,2) (0,1)
(M, M+1) (M-1, M+1) (M-1, M)

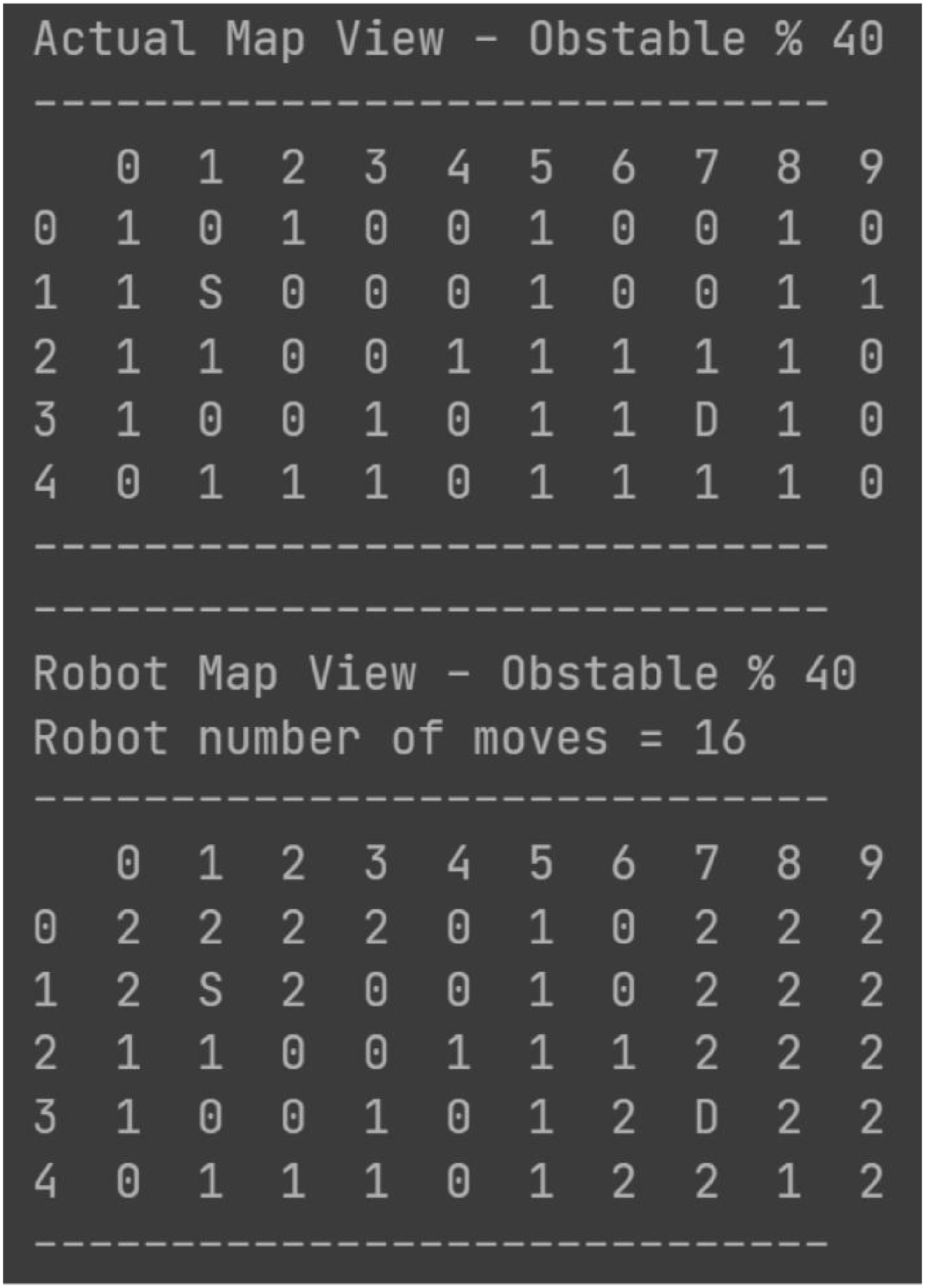
↘ (1,1) at 315°
270° 315° 0°
(2,1) (2,2) (1,2)
(M+1, M) (M+1, M+1) (M, M+1)



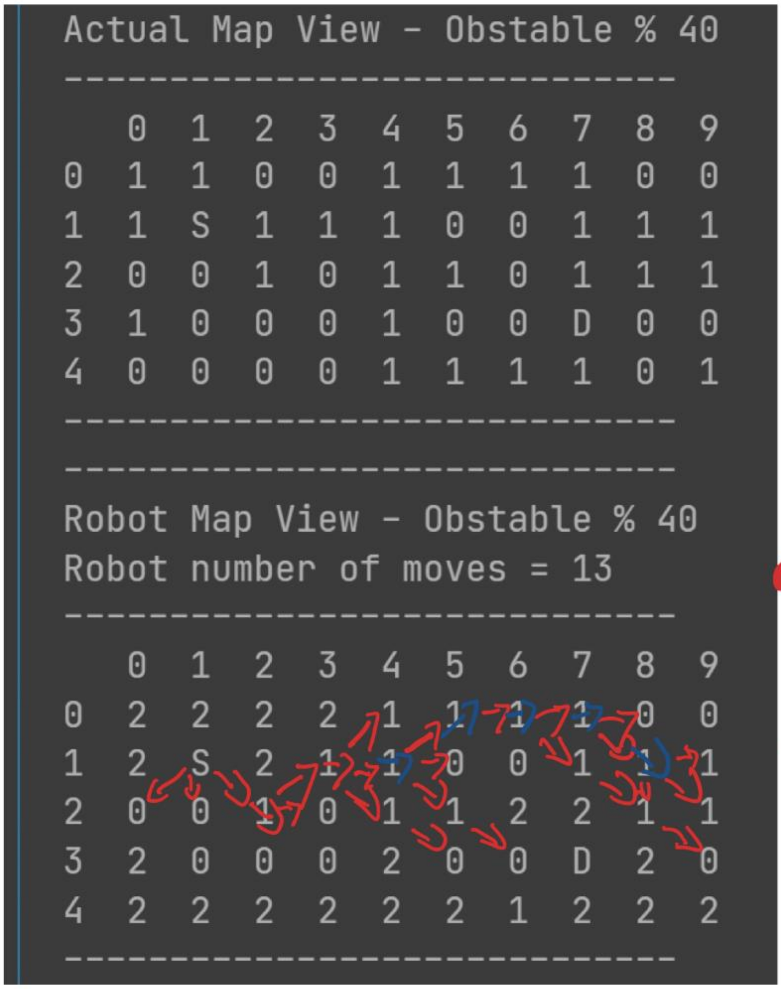
```
-----------------------------
Actual Map View - Obstable % 40
-----------------------------
   0  1  2  3  4  5  6  7  8  9
0  1  1  0  0  1  0  1  1  1  1
1  0  S  1  1  0  1  1  1  1  1
2  1  1  0  0  0  1  1  0  0  0
3  1  0  0  0  1  0  0  D  0  0
4  1  0  1  1  1  0  0  1  0  1
-----------------------------

-----------------------------
Robot Map View - Obstable % 40
Robot did not find a solution
-----------------------------
   0  1  2  3  4  5  6  7  8  9
0  1  1  0  0  1  0  2  2  2  2
1  0  S  1  1  0  2  2  2  2  2
2  1  1  0  0  0  2  2  2  2  2
3  1  0  0  0  2  2  2  D  2  2
4  1  0  2  2  2  2  2  2  2  2
-----------------------------

-----------------------------
Robot Map Moves View - Obstable % 40
Robot did not find a solution
-----------------------------
    0   1   2   3  4  5  6   7  8  9
0   7  13  13  -1  7 -1  0   0  0  0
1  -1   S  13  14 -1  0  0   0  0  0
2  13  13   1  -1 -1  0  0  -1 -1 -1
3  19  -1  -1  -1  0 -1 -1   D -1 -1
4  10  -1   0   0  0 -1 -1   0 -1  0
-----------------------------
```



```
Actual Map View - Obstable % 40
------------------------------
   0  1  2  3  4  5  6  7  8  9
0  0  0  0  1  1  1  0  1  1  0
1  0  S  1  0  1  0  1  1  1  1
2  0  1  1  1  1  1  0  0  0  0
3  0  1  0  1  0  1  1  D  0  1
4  0  1  1  1  0  0  1  1  0  1
------------------------------

------------------------------
Robot Map View - Obstable % 40
Robot number of moves = 200
------------------------------
   0  1  2  3  4  5  6  7  8  9
0  0  0  0  2  2  1  2  2  2  2
1  0  S  1  0  1  0  2  2  2  2
2  0  1  1  1  1  2  2  2  2  2
3  0  1  0  1  0  2  2  D  2  2
4  0  1  1  1  0  2  2  2  2  2
------------------------------
```

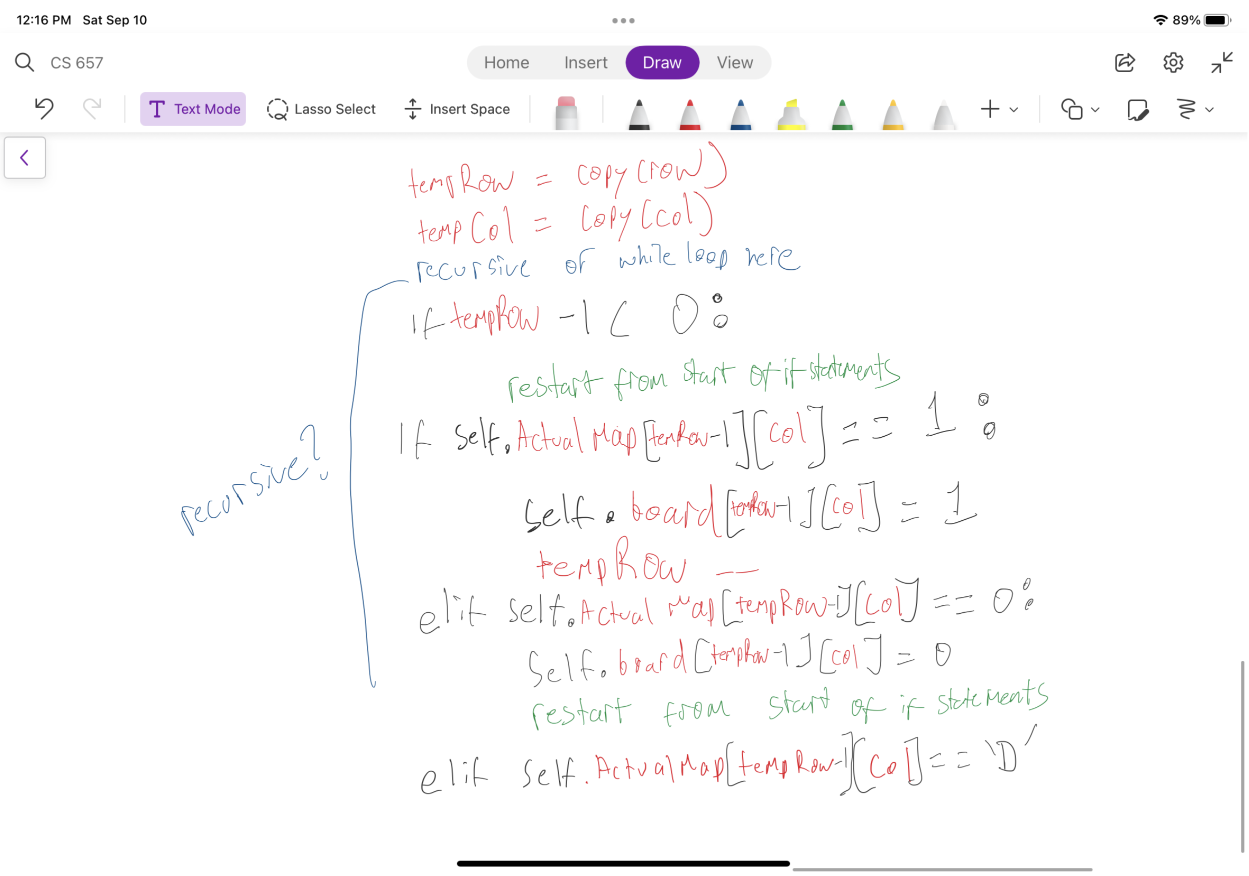


```
Actual Map View - Obstable % 40
-----------------------------
   0  1  2  3  4  5  6  7  8  9
0  1  0  1  0  0  1  0  0  1  0
1  1  S  0  0  0  1  0  0  1  1
2  1  1  0  0  1  1  1  1  1  0
3  1  0  0  1  0  1  1  D  1  0
4  0  1  1  1  0  1  1  1  1  0
-----------------------------
-----------------------------
Robot Map View - Obstable % 40
Robot number of moves = 16
-----------------------------
   0  1  2  3  4  5  6  7  8  9
0  2  2  2  2  0  1  0  2  2  2
1  2  S  2  0  0  1  0  2  2  2
2  1  1  0  0  1  1  1  2  2  2
3  1  0  0  1  0  1  2  D  2  2
4  0  1  1  1  0  1  2  2  1  2
-----------------------------
```



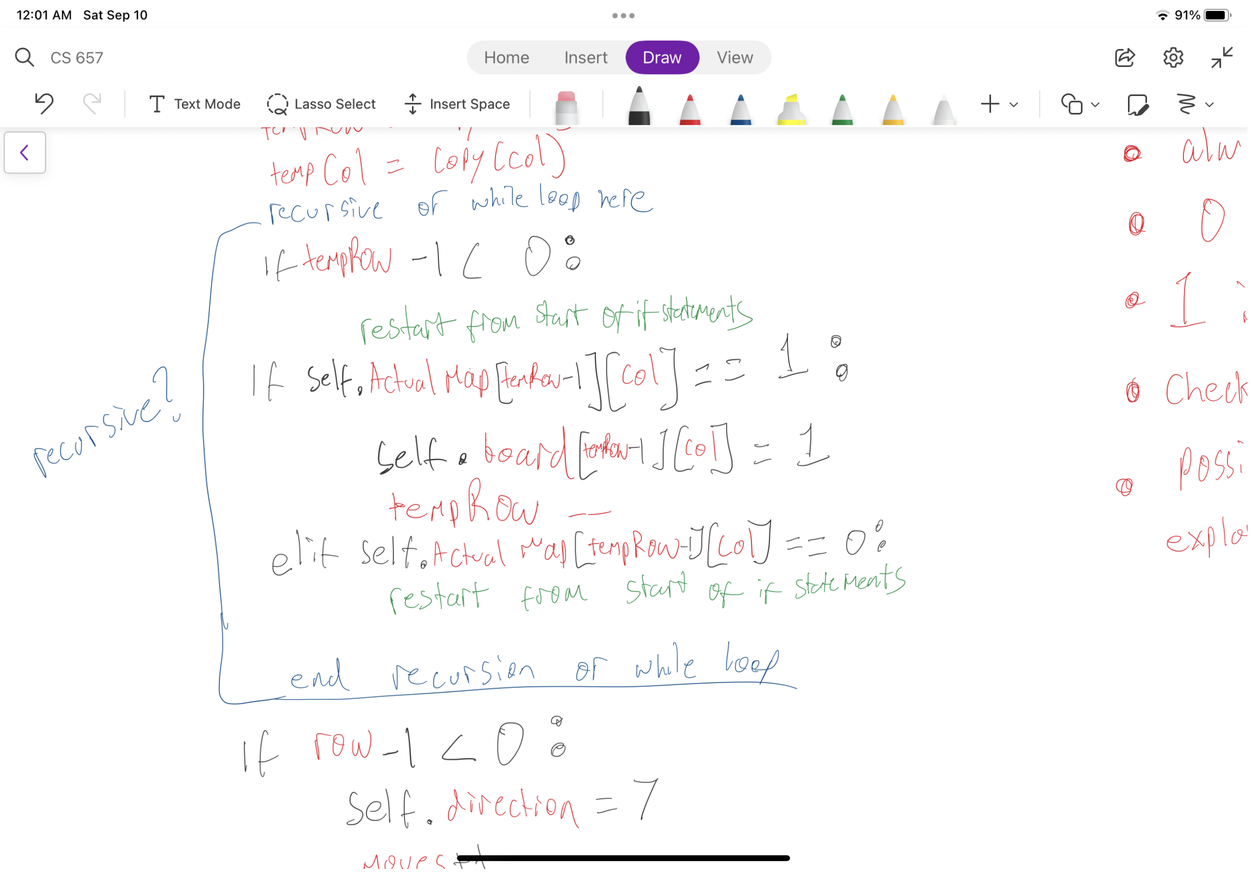
```
Actual Map View - Obstable % 40
-----------------------------
   0  1  2  3  4  5  6  7  8  9
0  1  1  0  0  1  1  1  1  0  0
1  1  S  1  1  1  0  0  1  1  1
2  0  0  1  0  1  1  0  1  1  1
3  1  0  0  0  1  0  0  D  0  0
4  0  0  0  0  1  1  1  1  0  1
-----------------------------

-----------------------------
Robot Map View - Obstable % 40
Robot number of moves = 13
-----------------------------
   0  1  2  3  4  5  6  7  8  9
0  2  2  2  2  1  1  1  1  0  0
1  2  S  2  1  1  0  0  1  1  1
2  0  0  1  0  1  1  2  2  1  1
3  2  0  0  0  2  0  0  D  2  0
4  2  2  2  2  2  2  1  2  2  2
-----------------------------
```

- direction 1
  didnt look at North=0
- direction 3
  didnt look



```
tempRow = copy(row)
temp Col = Copy(col)
recursive of while loop here
if tempRow -1 < 0:
        restart from start of if statements
If self.ActualMap[temRow-1][col] == 1:
        self.board[temRow-1][col] = 1
        tempRow --
elif self.Actual map[tempRow-1][col] == 0:
        self.board[tempRow-1][col] = 0
        restart from start of if statements
elif self.ActualMap[tempRow-1][col] == 'D'
```

recursive?



```
temp Col = copy(col)
recursive of while loop here
if tempRow -1 < 0:
        restart from start of if statements
if self.ActualMap[tempRow-1][col] == 1:
        self.board[tempRow-1][col] = 1
        tempRow --
 elif self.Actual map[tempRow-1][col] == 0:
        restart from start of if statements

 end recursion of while loop
if row-1 < 0:
        self.direction = 7
        moves
```

recursive?

- alw
- 0
- 1
- Check
- possi
- explo

V. Conclusion

VI. Overall it was interesting to code this program because the coding part was not hard just coming up with good rules that can provide the best solution, which I didn't feel I made because I found many boards were my program was not able to find a solution, despite me manually checking and seeing solutions, I identified some of this cases but wasn't able to make more rules due to time, but overall its working solution for most cases, also I did not implement any other searching algorithm.