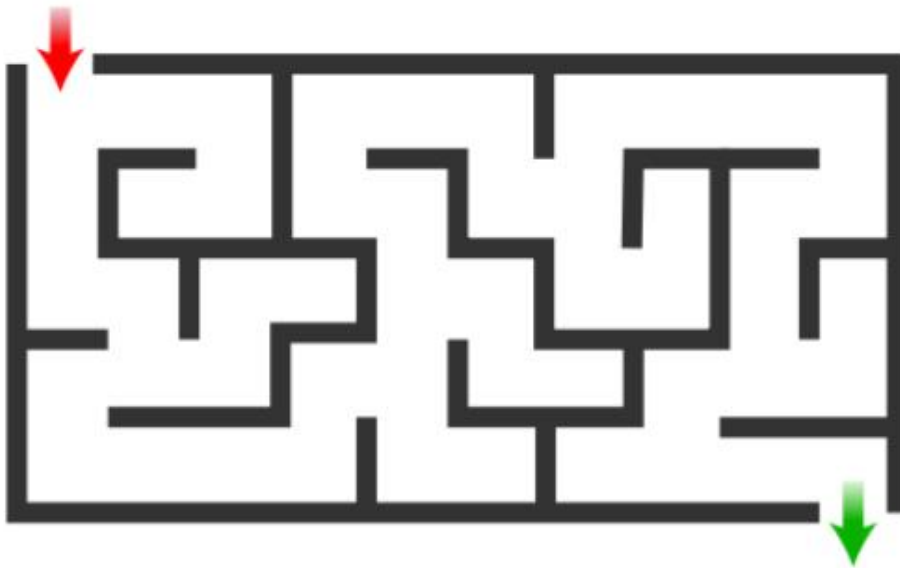


Programming Assignment

Diamonds Maze

Part1 (10 marks)

Overview: A maze is a tour puzzle in the form of a complex branching passage through which the solver must find a route. The pathways and walls in a maze are .fixed



The goal is to choose a path to the exit of the maze

For example if the input is a list with "." meaning empty space and "-" means blocked like this:

```
[ . / . / - / - / - / - / - / - / - / - /  
- / . / - / - / - / - / - / - / - / - /  
- / . / - / . / - / - / - / - / - / - /  
- / . / - / - / - / - / - / - / - / - /  
- / . / - / . / - / - / - / - / - / - /  
- / - / - / - / - / - / - / - / - / - / ]
```

The solution would be

Solution =

```
X  X  -  -  -  -  -  -  -  -
-  X  -  -  .  .  .  .  -  -
-  X  -  -  -  -  -  -  -  -
-  X  X  X  -  -  X  X  X  -
-  .  -  X  -  -  X  -  X  -
-  .  -  X  X  X  X  -  X  -
-  .  -  .  -  -  -  -  X  X
-  .  .  .  -  -  -  -  -  -
-  -  -  -  -  -  -  -  -
```

- 1- Define a suitable data structure **(1 marks)**
- 2- Define the moves that generate the new states using prolog **(5 marks)**.
- 3- Implement Depth first search to find a solution for this problem using prolog **(2 marks)**.
- 4- Print the valid path on the given maze board at the end of the program **(1 marks)**.
- 5- Print the number of visited nodes **(1 marks)**.

In order to do that you will use function "Expand nodes" as you know expand nodes calls all the moves and puts the new children in a list. When putting the new children in a list put them by calling move_up, move_down, move_left, move_right in this order. That would easier for debugging when discussing the assignment.

Part 2 (10 marks)

Assume that the maze contains diamonds. We want to implement the A* algorithm in order to find a path till the exit of the maze while at the same time maximizing the number of collected diamonds that is within the path

```
[ .  .  .  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  .  D  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  .  .  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  D  D  D  -  -  -  .  D  .  -  -  -  -  -  -  -  -  -  -  -  -
-  .  -  -  .  -  D  D  D  -  D  -  -  -  -  -  -  -  -  -  -  -
-  .  -  -  .  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  .  -  -  D  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - ]
```

If this is the input list that represents the maze where "." means empty "-" means blocked and "D" is a diamond

The optimal path that will be found by the A*

X	X	-	-	-	-	-	-	-	-
-	X	-	-	-	-
-	X	-	-	-	-	-	-	-	-
-	X	X	X	-	-	X	X	X	-
-	.	-	X	-	-	X	-	X	-
-	.	-	X	X	X	X	-	X	-
-	.	-	.	-	-	-	-	X	X
-	D	.	D	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-

Maximum number of collected diamonds=9

- 1- You will implement the search algorithm A*.
- 2- define the heuristic in order to choose the path with the maximum number of collected diamonds **(6 marks)**.
- 3- Print the number of collected diamonds **(2 mark)**.
- 4- Print the number of visited nodes **(2 marks)**.

Rules of submission:

1. Students can form teams of 3 to work on this assignment.
2. Students can only use Prolog to implement this assignment.
3. No late submissions.
4. Each part should be implemented in a separate file
5. Students required to add comments to explain their code.