

Name : **Ritik Mandloi**
Roll No: **180101066**

PLL Assignment VI
Readme

I have provided the following files in the zip folder.

1. 'generateMaze 2.cpp'
2. 180101066.pl

Prerequisites :

Install graphviz by using command:

`sudo apt-get install graphviz`

Install swi-prolog by using command:

`sudo apt-get install swi-prolog`

INSTRUCTIONS FOR EXECUTION:

1. Compile the cpp file :

`g++ 'generateMaze 2.cpp' -o generateMaze`

2. Execute using the below format :

`./generateMaze Height:int Width:int FaultProbability:fp<0,1> SrcX SrcY DstX DstY`

Example: `./generateMaze 10 10 0.3 1 1 9 9`

`./generateMaze 3 10 0.5 1 0 2 8`

It generate "Mazedata.pl" file with contain facts for prolog

It also generate graph1.png, which shows the connectivity of Maze

The source node and destination node

3. Append contents of 180101066.pl to Mazedata.pl :

`cat 180101066.pl >> Mazedata.pl`

4. Now execute the command: `swipl -s Mazedata.pl`

5. Now type the query in the format:

`shortestpath(SrcX,SrcY,DstX,DstY, Height, Width).`

Example: `shortestpath(1,0,2,8,3,10).`

`shortestpath(1,1,9,9,10,10).`

If the query is valid it prints the shortest path array along with its length.

6. You can add a faulty node in the query by typing:

`assertz(faultynode(NodeNumber)).`

Example: `assertz(faultynode(24)).`

`assertz(faultynode(15)).`

7. You can remove a faulty node in the query by typing:

`retract(faultynode(NodeNumber)).`

Example: `retract(faultynode(15)).`

`retract(faultynode(25)).`

Screenshots are attached for better understanding.

Compilation:

```
mandloi@mandloi-VirtualBox:~/Music/180101066_AssignVI$ g++ 'generateMaze 2.cpp' -o generateMaze
mandloi@mandloi-VirtualBox:~/Music/180101066_AssignVI$ ./generateMaze 3 10 0.5 1 0 2 8
mandloi@mandloi-VirtualBox:~/Music/180101066_AssignVI$ cat 180101066.pl >> Mazedata.pl
mandloi@mandloi-VirtualBox:~/Music/180101066_AssignVI$ swipl -s Mazedata.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- shortestpath(1,0,2,8,3,10).
No Path Exists Between The Given Nodes
true.
```

Queries:

```
mandloi@mandloi-VirtualBox:~/Music/180101066_AssignVI$ swipl -s Mazedata.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
```

```
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
```

```
?- shortestpath(1,0,2,8,3,10).
No Path Exists Between The Given Nodes
true.
```

```
?- shortestpath(1,0,0,2,3,10).
Shortest Path Length = 5
Shortest Path = [10,7,4,1,2]
true.
```

```
?- assertz(faultynode(1)).
true.
```

```
?- shortestpath(1,0,0,2,3,10).
No Path Exists Between The Given Nodes
true.
```

```
?- retract(faultynode(1)).
true.
```

```
?- shortestpath(1,0,0,2,3,10).
Shortest Path Length = 5
Shortest Path = [10,7,4,1,2]
true.
```

```
?- retract(faultynode(15)).
true.
```

```
?- retract(faultynode(25)).
true.
```

```
?- shortestpath(1,0,2,8,3,10).
Shortest Path Length = 9
Shortest Path = [10,9,12,15,18,21,24,25,28]
true.
```

```
?- assertz(faultynode(25)).
true.
```

```
?- shortestpath(1,0,2,8,3,10).
No Path Exists Between The Given Nodes
true.
```

