

# Assignment 04 (Due: Wednesday, November 18, 2020, 11 : 59 : 00PM Central Time)

CSCE 322

## Contents

<b>1</b>	<b>Instructions</b>	<b>1</b>
1.1	Data File Specification . . . . .	1
1.2	oddWidthEvenInversions(Puzzle) . . . . .	2
1.3	quickestWin(Puzzle,Moves) . . . . .	2
1.4	twoConditions(Puzzle) . . . . .	3
1.5	puzzleSolvable(Puzzle) (15% Extra Credit) . . . . .	3
<b>2</b>	<b>Compilation &amp; Execution</b>	<b>3</b>
<b>3</b>	<b>Webgrader Warning</b>	<b>3</b>
<b>4</b>	<b>Naming Conventions</b>	<b>4</b>
<b>5</b>	<b>Point Allocation</b>	<b>4</b>
<b>6</b>	<b>External Resources</b>	<b>4</b>

## List of Figures

### 1 Instructions

This assignment will use Prolog to extract certain information about the state of a **Boss puzzle**.

#### 1.1 Data File Specification

The puzzle will be represented as a list of lists of numbers.

```
[l,r,u,u,r,l,l,r,r,r,u,r,u].
[81,32,104,1,105,30,74,68,9].
[65,4,36,108,88,57,21,66,76].
[56,73,35,126,45,75,93,125,114].
[116,12,44,85,23,3,130,7,50].
[102,106,118,96,122,18,109,84,61].
[113,53,42,95,92,33,123,25,90].
[17,111,60,119,24,51,82,38,117].
[77,46,14,86,80,115,64,16,112].
[70,94,83,132,62,8,48,5,15].
[110,69,31,0,98,78,43,22,121].
[41,58,103,99,39,40,101,100,29].
[72,54,89,131,2,55,47,129,133].
[49,67,127,11,128,97,19,26,87].
[52,37,120,13,71,107,10,34,63].
[27,6,28,124,91,134,20,59,79].
```

## 1.2 oddWidthEvenInversions(Puzzle)

The query `oddWidthEvenInversions(Puzzle)` will be successful when the puzzle has an odd number of columns and an even number of inversions in the puzzle.

```
[115,110,124,77,1,99,17,18,40,53]
[42,89,24,5,91,10,72,75,81,60]
[132,119,118,0,48,37,13,84,68,41]
[23,85,44,127,135,141,107,139,64,67]
[80,138,130,149,38,4,126,144,6,49]
[102,86,26,78,123,79,34,98,74,3]
[106,147,21,9,146,145,2,120,92,36]
[47,19,136,31,142,11,30,114,20,112]
[121,39,143,45,93,15,100,108,109,33]
[66,111,70,29,73,25,14,69,32,96]
[97,104,87,82,76,117,105,131,46,90]
[8,22,148,43,88,113,56,116,65,62]
[63,52,133,140,94,54,35,71,103,55]
[128,101,28,122,16,95,83,137,59,12]
[61,57,125,58,129,51,7,27,50,134]
false
```

## 1.3 quickestWin(Puzzle,Moves)

The query `quickestWin(Puzzle,Moves)` will be successful when `Moves` is unified with the shortest valid path (fewest number of slides) from the current state of the puzzle to a solved puzzle. There may be multiple shortest paths, and `quickestWin(Puzzle,Moves)` should report them all, if ; appears in the query. No sequence should exceed 10 moves in length.

```
[1,2]
[3,0]
paths
[]
true
```

#### 1.4 twoConditions(Puzzle)

The query `twoConditions(Puzzle)` will be successful when the puzzle has an odd width and even number of inversions, or the puzzle has an even width and an odd number of inversions.

```
[36,64,1,20,52,77,6,35,10,83,31,5]
[3,16,21,62,2,81,50,30,67,78,45,49]
[27,43,32,80,24,46,14,73,70,7,18,79]
[55,19,72,34,71,76,4,42,58,37,11,68]
[28,25,56,74,63,15,29,8,51,39,44,23]
[66,41,9,82,69,26,59,12,54,17,47,13]
[38,48,75,57,40,60,22,53,33,0,61,65]
true
```

#### 1.5 puzzleSolvable(Puzzle) (15% Extra Credit)

The query `puzzleSolvable(Puzzle)` will be successful when the puzzle is solvable (according to <http://www.cs.bham.ac.uk/~mdr/teaching/modules04/java2/TilesSolvability.html>).

```
[37,85,56,24,59,16,83,28,20,10,9,69,25,62,79]
[84,65,22,54,31,17,18,21,61,50,0,19,55,60,76]
[32,3,42,26,38,46,58,29,51,2,71,7,30,27,44]
[72,67,13,12,4,23,75,88,52,74,73,89,6,14,5]
[86,1,87,47,66,35,82,40,45,63,43,49,48,15,41]
[81,33,8,53,80,34,78,77,64,36,68,11,70,57,39]
true
```

## 2 Compilation & Execution

Your program will be tested on `cse.unl.edu`, using pl. `hmrk04tests.pl` will include test cases for testing your program. You can run the test cases with the commands:

```
[hmrk04tests].
loadHelpers.
part01.
```

## 3 Webgrader Warning

Prolog programs can take some time to run. The webgrader is limiting individual processes to 5 minutes. The CSE server is currently running version 7.6.4 of SWI-Prolog.

## 4 Naming Conventions

You will be submitting at least 3 .pl files. The filenames should be `csce322hmwrk04prt01.pl`, `csce322hmwrk04prt02.pl` and `csce322hmwrk04prt03.pl`

## 5 Point Allocation

Component	Points
<code>csce322hmwrk04prt01.pl</code>	
Test Cases	$35 \times 1$
Total	35
<code>csce322hmwrk04prt02.pl</code>	
Test Cases	$35 \times 1$
Total	35
<code>csce322hmwrk04prt03.pl</code>	
Test Cases	$30 \times 1$
Total	30
Total	100

## 6 External Resources

[Prolog - Wikibooks](#)

[Learn Prolog Now!](#)

[Prolog Tutorial Category:Prolog - Rosetta Code](#)