COMPSCI 761 Assignment 2 due Friday 30 September 2022

Due: **by 11:59pm on the due date**. Late assignments will not be marked.

Total mark: **10**.

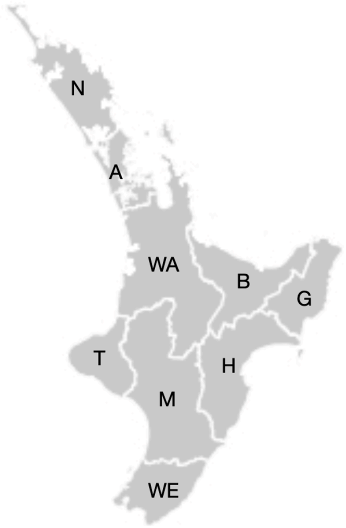
Worth: This assignment counts towards 10% of your final mark.

1. [**3 marks**] Solve the map colouring problem for North Island using Prolog. Up to three colours, `red`, `green`, `blue`, are allowed to use. Two adjacent regions cannot be marked with the same colour.

You may use the following knowledge base:

- `different(red, green).`
- `different(red, blue).`
- `different(green, blue).`
- `different(green, red).`
- `different(blue, red).`
- `different(blue, green).`

Implement a Prolog predicate `mapcolouring(N,A,WA,B,G,T,M,H,WE)` where a variable should be the corresponding word as indicated in the following map. Submit your answer as `mapcolouring.pl` file.



2. [**4 marks**] The *Legend of Zelda* is an epic fantasy serial of action-adventure video games. It has been worldwide renowned for more than 35 years. The story happens in an ancient Kingdom called Hyrule. The series of games centre on the various incarnations of *Link*, a courageous knight of the kingdom, and *Zelda*, a magical princess of the kingdom. They fight together to save Hyrule from *Ganon*, a devil that wants to destroy the kingdom. At the end of each game of the series, Link uses a legendary sword, *master sword*, to defeat and seal the devil Ganon and successfully save the kingdom.

Now, consider the following scenario: princess Zelda is locked up by Ganon in a castle. To save Zelda, Link must beat Ganon and open the castle's gate. But he cannot beat Galon without the master sword. This scenario is abstracted into a 3×4 grid world. Initially, Link is at $(1, 1)$, master sword at $(3, 1)$, Ganon at $(2, 2)$, and three gates at $(1, 3)$, $(2, 3)$ and $(3, 3)$. Each time step, Link can take an action from the following options: *Move* to an adjacent grid, *Pick* up the master sword, *Attack* Ganon, and *Open* the gate. The result of a *Pick* is that Link holds the master sword if he is at the same grid as master sword (the grid where the master sword is placed becomes empty after a successful *Pick*). The result of an *Attack* is that Link defeats Ganon if he holds the master sword and is at the same grid as Ganon (the grid where Ganon is located becomes empty after a successful *Attack*). The result of an *Open* is that the corresponding gate is opened if Ganon is defeated and Link is at the same grid as the gate.

Link will die if he moves to Ganon's grid without the master sword. When Link is adjacent to Ganon, he can feel an atmosphere of danger, called *Calamity*. He can also sensor the *Aura* of the master sword if he is one step toward it.

We use the following atomic propositions to generate a knowledge base:

For any $i \in \{1, 2, 3\}$ and $j \in \{1, 2, 3, 4\}$

- $C_{i,j}$: Calamity can be sensed at location (i, j)
- $G_{i,j}$: Ganon is at location (i, j)
- $S_{i,j}$: Master sword is at location (i, j)
- $HoldingSword$: Link has picked up the master sword

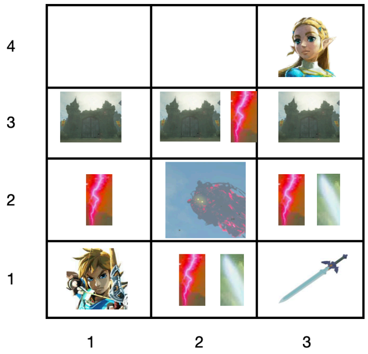
(a) Use atomic propositions given above to represent the following static knowledge axioms:

(1) Ganon must be at one location.

(2) Ganon and master sword are not at the same location.

(3) Calamity cannot be sensed at two adjacent locations at the same time.

(b) Use atomic propositions given above to define dynamic knowledge axiom $OK_{i,j}$ that denotes that it is safe to move to location (i, j).

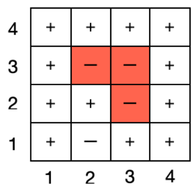


Submit your answer as `zelda.pdf` file. (*You are asked to write a logic formula to represent or define each axiom. The explanation for how you derived the formula is not required.*)

3. [**3 marks**] Given is a 4×4 board with each grid marked with either "+" or "−". The goal is to find all − grids that are *surrounded* by +. A − grid at the edge *cannot* be surrounded. More formally, each grid can have up to 4 neighbour grids, i.e., top, bottom, left and right. If one grid or a group of "−" grids forms a "connected" area and their neighbours are all "+" grids, then all "−" grids in this group are *surrounded*. Otherwise, if any a − grid in that group has a − neighbour that is at the edge, then all neg grids in this group are non-surrounded. For example, in figure (a), the − grids highlighted in red are surrounded; in figure (b), no − grid is surrounded.

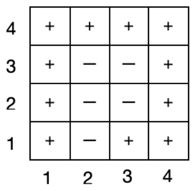
Use the constants g11, g12, . . . , g44 to denote the coordinates of grids. Use the rules `neg`(G) and `pos`(G) to denote a grid marked with − and +, respectively, where the variable G ∈ {g11, g12, . . . , g44}.

You are requested to: Build a knowledge base and implement a Prolog predicate `is_surrounded` (G) such that `is_surrounded` (G) is true if G is a − grid and is surrounded by + grids, and false otherwise. You are allowed to define any additional and auxiliary rules.



4	+	+	+	+
3	+	−	−	+
2	+	+	−	+
1	+	−	+	+
	1	2	3	4

(a)



4	+	+	+	+
3	+	−	−	+
2	+	−	−	+
1	+	−	+	+
	1	2	3	4

(b)

Evaluation. Your programme may be tested using different configurations of the board. For example, the configuration code for the figure (a) is:

```
neg(g21).
neg(g23).
neg(g32).
neg(g33).


pos(g11).
pos(g12).
pos(g13).
pos(g14).
pos(g22).
pos(g24).
pos(g31).
pos(g34).
pos(g41).
pos(g42).
pos(g43).
pos(g44).
```

The configuration code will be added at the top of the knowledge base when testing.

Submit your answer as `surround.pl` file.