

## AI Prolog data structure exercise 2025

Given that the code provided here constructing a grid with values in it. It does this using the assert and retract predicates. To maintain states using lists we would like to have a more dynamic construction in memory.

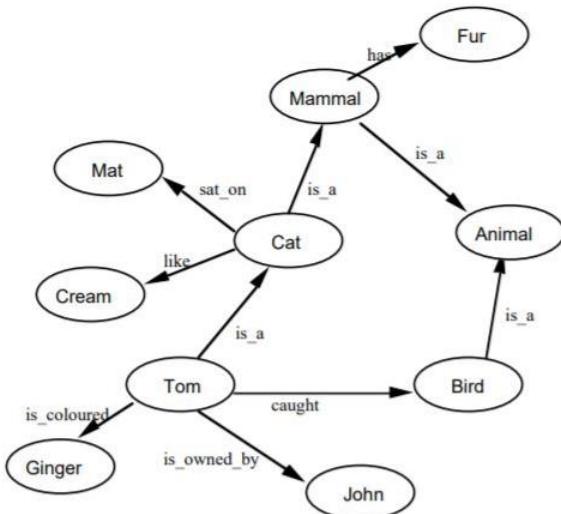
The following code produces a list of numbers counting down from N to 0 and inserts them in reverse order.

```
number_list([], 0).
number_list([N|T], N) :-  
    N1 is N - 1,  
    number_list(T, N1).
```

1. Construct a Prolog predicate that will return a list of locations based on an  $N \times N$  grid.  
Possible formats for the initial empty list could be
  - a.  $[[1,1, e], \dots, [N, N, e]]$  or
  - b.  $[[1,1, []], \dots, [N, N, []]]$
2. Use Prolog list manipulation predicates documented [here](#) to place a few items at a specified location in a list constructed in 1.
3. Write a predicate that will print out the grid in order padding out the grid locations to allow for the cell with the most items in it see the format print command [here](#).
4. Write an adaption of the adjacency to return the Adjacent locations to the current location in the List data structure presented in 3 above.

## Semantic network representations

Based on the sample code provided write a set of triples that will support the following semantic network. Can you construct a rule that will prove Tom caught an animal?



## Frame representation Exercise

Using the Frame example code provided and the fact that Martha Rother is the Project leader for the project ‘Nano technology transfer’ Use the fill frame rule (amend if necessary) that will make a new nano technology transfer technical report.

## XPCE UI exercise

SWI Prolog comes with an object-oriented programming language called XPCE. Part of this object-oriented language allows you to create UIs. See the Grid box example to see how you can build a GUI for a grid.

Adopt the code provided to present the list of cells on a GUI built using XPCE.