

Lab 2

Exercise 1

Try the following in Prolog. Note that atom/1, atomic/1 and functor/3 are built in Prolog predicates.

```
11 ?- atom(dog).  
true.  
  
12 ?- atom(man(marcus, 40)).  
false.  
  
13 ?- atom(X).  
false.  
  
14 ?- atom(23).  
false.  
  
15 ?- atomic(23).  
true.  
  
16 ?- atomic(man(marcus, 40)).  
false.  
  
17 ?- functor(man(marcus, 40)).  
ERROR: Undefined procedure: functor/1  
ERROR:         However, there are definitions for:  
ERROR:         functor/3  
false.  
  
18 ?- functor(man(marcus, 40), A, B).  
A = man,  
B = 2.  
  
19 ?- atomic([1,2,3]).  
false.
```

Numeric exercises

Exercise 2

Try the following in Prolog interpreter:

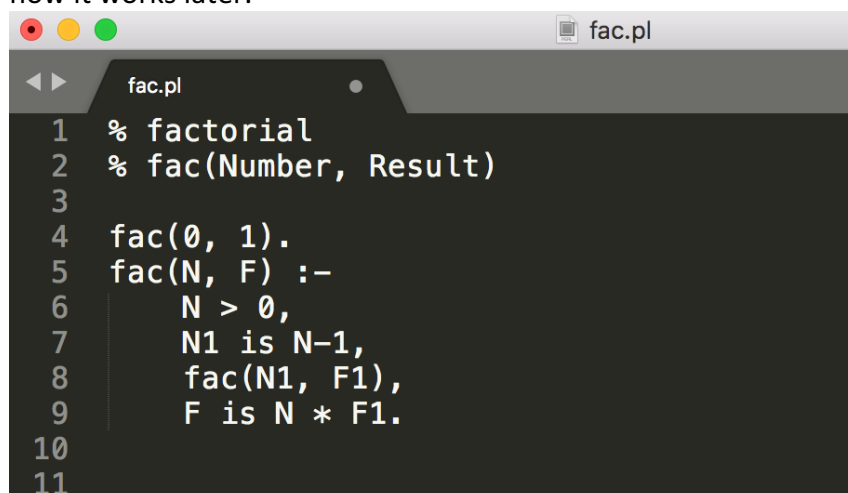
```
?- X is 5 mod 2.  
?- X = 5 mod 2.  
?- Y is mod(7,3).  
?- Y = mod(7, 3).  
?- number(9).  
?- number(a9).  
?- number(3.456).
```

Exercise 3 - try the following

```
?- X = 4.  
X = 4.  
  
?- X == 4.  
false.  
  
?- X = 1+3, X = 4.  
false.  
  
?- X = 1+3, X == 4.  
false.  
  
?- X is 1+3, X == 4.  
X = 4.  
  
?- X is 1+3, X = 4.  
X = 4.  
  
?- 2 \= 3.  
true.  
  
?- X \= 3.  
false.  
  
?- U is 5 mod 2.  
U = 1.  
  
?- U is 5 div 2.  
U = 2.  
  
?- U is 5 / 2.  
U = 2.5.  
  
?- |
```

Exercise 4

Used a text editor to write the following code, [fac.pl](#), for calculating the factorial of a number. This shows you how to do simple recursion using predicates in Prolog. I will explain how it works later.



```
1  % factorial  
2  % fac(Number, Result)  
3  
4  fac(0, 1).  
5  fac(N, F) :-  
6      N > 0,  
7      N1 is N-1,  
8      fac(N1, F1),  
9      F is N * F1.  
10  
11
```

Load into Prolog and run it as in:

```
% /Users/richard/Dropbox/my Prolog/fac.pl compiled 0.00 sec, 10
clauses
?- fac(3, R).
R = 6
false.

?- fac(3, R).
R = 6 .

?- fac(8, R).
R = 40320 .

?- fac(1, R).
R = 1 .

?-
```

Exercise 5

Can you write a similar program to calculate the Fibonacci series 0,1,1,2,3,5,8,13,21,34 ...?

E.g. `fib(3,R)` gives `R = 2`. `fib(6,Y)` gives `Y = 8`.

Recall $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$.