****

**Faculté de Génie – EECS**

**CSI2520 : PARADIGMES DE PROGRAMMATION**

**Hiver 2015 – Tutorat 4**

**Exercice 1**

Que fait ce prédicat:

premier([X|\_],X).

Quel est le résultat de la requête ?

?- premier([2,4,6],2).

Et celui-ci:

dernier([X],X).

dernier([\_|L],X) :- dernier(L,X).

Et enfin celui-là

compte([],0).

compte([\_|R],N) :- compte(R,N1), N is N1+1, N>0.

**Exercice 2**

Donnez la définition du prédicat occurrence (L,X,N) qui est vrai si N est le nombre de fois où X est présent dans la liste L.

**Exercice 3**

Soit le programme suivant:

main :-

open('c:/fruit.txt', read, Str),

read\_file(Str,Lines),

close(Str),

write(Lines), nl.

read\_file(Stream,[]) :-

at\_end\_of\_stream(Stream).

read\_file(Stream,[X|L]) :-

\+ at\_end\_of\_stream(Stream),

read(Stream,X),

read\_file(Stream,L).

Quel est le résultat de la requête :

?- main.

Si le fichier fruit.txt contient les données suivantes:

pomme.

tomate.

orange.

celeri.

poire.

salade.

**Exercice 4**

Comment peut-on définir le tri par insertion

tri\_insertion(L,L1) : qui construit la liste triée L1 des éléments de L ?

**Exercice 5**

Comment peut-on définir le prédicat *permut* qui retourne toutes les permutations possibles d’une liste.

Voici un exemple d’utilisation :

/\* Exemple :   
?- permut([1,2,3],L).   
L = [1,2,3] ;   
L = [1,3,2] ;   
L = [2,1,3] ;   
L = [2,3,1] ;   
L = [3,1,2] ;   
L = [3,2,1] ;   
no   
\*/