TD de programmation fonctionnelle et logique
Corrigé du TD 8 : premiers programmes Prolog

1. Écrire un prédicat « hypoténuse » qui prend trois arguments X, Y et Z et qui est vrai si $Z^2 = X^2 + Y^2$ (l'égalité entre expressions arithmétiques se note « =:= »).

   ```
   hypoténuse(X,Y,Z) :- X*X + Y*Y =:= Z*Z.
   ```

2. Écrire un prédicat qui prend en entrée un élément et une liste et qui est vrai si l'élément appartient à la liste.

   ```
   estelt(X,[X|Y]).
   estelt(X,[Y|L]) :- estelt(X,L).
   ```

3. Écrire un prédicat qui est vrai si son argument est un chiffre.

   ```
   chiffre(T) :- estelt(T,[0,1,2,3,4,5,6,7,8,9]).
   ```

4. Réécrire le prédicat « hypoténuse » pour qu'il puisse énumérer tous les triplets (X, Y, Z) de chiffres satisfaisants la relation $X^2 + Y^2 = Z^2$.

   ```
   hypoténuse(X,Y,Z) :- chiffre(X), chiffre(Y), chiffre(Z),
    X*X + Y*Y =:= Z*Z.
   ```

5. Écrire un prédicat qui est vrai si tous les éléments d'une liste sont des chiffres.

   ```
   tous_chiffres([]).
   tous_chiffres([T|L]) :- estelt(T,[0,1,2,3,4,5,6,7,8,9]), tous_chiffres(L).
   ```

   Autre solution :

   ```
   tous_chiffres([]).
   tous_chiffres([T|L]) :- between(0,9,T), tous_chiffres(L).
   ```

6. Écrire un prédicat prenant un élément et une liste en argument, et qui est vrai si et seulement si l'élément n'appartient pas à la liste (le symbole différent est « =\= »).

   ```
   paselt(X,[]).
   paselt(X,[Y|L]) :- paselt(X,L), X =\= Y.
   ```

7. Écrire un prédicat qui est vrai si tous les éléments d'une liste sont (deux à deux) différents.

   ```
   tous_différents([]).
   tous_différents([T|L]) :- paselt(T,L), tous_différents(L).
   ```

8. Résoudre le problème :

   ```
      S E N D
   +  M O R E
   -----------
    M O N E Y
   ```

   où `S`, `E`, `N`, `D`, `M`, `O`, `R` et `Y` sont des chiffres deux à deux distincts.

   ```
   somme([S,E,N,D,M,O,R,Y]) :-
      tous_chiffres([S,E,N,D,M,O,R,Y]),
      tous_différents([S,E,N,D,M,O,R,Y]),
      ((D + 10*N + 100*E + 1000*S) + (E + 10*R + 100*O + 1000*M)) =:=
        (Y + 10*E + 100*N + 1000*O + 10000*M).
   ```

9. Écrire un prédicat qui prend en entrée un élément et deux listes et qui est vrai si la deuxième liste peut être obtenue par une quelconque insertion de l'élément dans la première liste.

```
insère(T,M,[T|M]):-
insère(T,[M|N],[M|Q]):- insère(T,N,Q).
```

10. Écrire un prédicat qui prend en entrée deux listes et qui est vrai si une liste est permutation des éléments de l'autre liste.

```
permutation([],[]).
permutation([T|R],L) :- permutation(R,M), insère(T,M,L).
```

11. Écrire une nouvelle solution du problème

```
   S E N D
+  M O R E
-----------
 M O N E Y
```

avec le prédicat sur les permutations.

```
tous_chiffres_différents(L) :- permutation([0,2,3,4,5,6,7,8,9],[X|[Y|L]]).
somme([S,E,N,D,M,O,R,Y]) :-
   tous_chiffres_différents([S,E,N,D,O,R,Y]), M = 1,
   ((D + 10*N + 100*E + 1000*S) + (E + 10*R + 100*O + 1000*M)) =:=
     (Y + 10*E + 100*N + 1000*O + 10000*M).
```