

TD 1



Exercice 1 :

Décrire ce qui se passe quand les expressions suivantes sont évaluées :

1. $(+ (- 5 1) (+ 3 7))$
2. $(\text{list } 1 (+ 2 3))$
3. $(\text{if } (\text{listp } 1) (+ 1 2) (+ 3 4))$
4. $(\text{list } (\text{and } (\text{listp } 3) t) (+ 1 2))$



Exercice 2 :

En utilisant `car` et `cdr`, écrire une fonction qui renvoie le quatrième élément d'une liste.



Exercice 3 :

Écrire une fonction qui prend deux arguments et renvoie le plus grand des deux.



Exercice 4 :

Écrire une fonction qui prend une liste comme argument et renvoie vrai si l'un de ses éléments est une liste.



Exercice 5 :

Écrire une version itérative puis récursive d'une fonction qui :

- (a) prend un entier positif et imprime autant de points
- (b) prend une liste et renvoie le nombre de fois le symbole `'a'` apparaît dans cette liste.



Exercice 6 :

Que font les deux fonctions suivantes :

```
(defun enigme(X)
  (and (not (null X))
        (or (null (car X))
              (enigme (cdr X)))))
```



```
(defun mystere (X Y)
  (if (null Y)
      nil
      (if (eql (car Y) X)
          0
          (let ((Z (mystere X (cdr Y))))
              (and Z (+ Z 1)))))))
```

Exercice 7 :

Un ami tente d'écrire une fonction qui renvoie la somme de tous les éléments non-nuls d'une liste. Il a deux versions de cette fonction, mais aucune d'elles n'est correcte. Dites pourquoi et proposer une version correcte.

```
(defun summit (liste)
  (remove nil liste)
  (apply #' + liste))
```



```
(defun summit (liste)
  (let ((X (car liste)))
    (if (null X)
        (summit (cdr liste))
        (+ X (summit (cdr liste))))))
```