

Q1. Soit l'expression binaire préfixée comme suit : $* - a * b c + \backslash d e - f g$
Donnez l'arbre syntaxique, les expressions : post-fixée et symétrique.

Q2. On tape l'expression suivante à scheme (à racket) : `(1 2 3 4)`
Quel est le résultat obtenu ?

Q3. Formulez les fonctions booléennes (**nand** *a b*) et (**nor** *a b*) sans utiliser les fonctions (**and** *a b*) (**or** *a b*) et (**not** *a*) définies dans scheme.

Q4. On connaît la fonction factorielle : $Fact(n) = 1 * 2 * \dots * (n-1) * n$
Donnez la définition (récursive) de cette fonction en scheme.

Les questions Q5 et Q8 vont traiter les représentations en octal et en binaire.
Supposons un nombre *N* représenté en base de 8 (chaque chiffre de *N* est entre 0 et 7, 0 et 7 compris).
Pour cette base, on peut utiliser les nombres (les caractères, les chiffres) décimaux. Par exemple :
N = 2017633 en base de 8 : `(define N 2017633)`

Q5. Écrivez la fonction (**octal?** *N*) qui retourne vrai (**#t**) si son argument est un nombre qui peut être interprété en octal (comme *N* ici présenté) et faux (**#f**) autrement. (Attention : la chaîne 11 en octal représente 9 en décimal.)

Q6. Écrivez la fonction (**valeur** *N*) qui retourne la valeur décimale de l'entier *N* codé en octal. Si *N* n'est pas un entier positif, elle donne un message d'erreur. Exemple :
`(valeur 2617) → 142310 (= 2*8*8*8 + 6*8*8 + 1*8 + 7)`

Q7. Soit *C* un chiffre en base de 8. Écrivez la fonction (**bits** *C*) qui retourne la valeur en binaire (des chiffres binaires correspondants). Par exemple, pour *C = 5*, cette fonction doit retourner 101. Attention, ici, on ne parle pas de listes.

Q8. On va reprendre le nombre *N* codé en octal représenté cette fois-ci avec **une liste** de ses chiffres. Par exemple pour *N = 2017633 en base de 8* : `(define N '(2 0 1 7 6 3 3))`. Écrivez la fonction (**binaire** *N*) qui retourne la valeur binaire correspondante sous forme d'une liste de bits. Rappel : une représentation octale peut être convertie en binaire position par position. Par exemple, soit (2 5) la représentation de la valeur en octal. Puisque le code binaire de 2 en octal est (0 1 0) et le code de 5 en octal est (1 0 1), la fonction doit retourner (0 1 0 1 0 1) (c'est la représentation binaire de la valeur en question).

Q9. Soit *L* une *liste de nombres* qui contient des nombres et des listes de nombres. Par exemple :
`(define L '(2 1 (4 (5 8)) ((7 6 (0) 6)) 3 (1)))`
Écrivez la fonction (**somme** *L*) qui additionne les éléments d'une liste de nombres quelque soit le niveau des éléments.

Q10. Un nombre premier est un nombre (positif) qui n'est divisible que par 1 et par lui-même.
Donnez la formulation itérative (et pas récursive) du prédicat (**premier?** *N*) qui retourne **#t** si *N* est un nombre premier.

Annexe

Iterations
(do

```
((var1 base1 exp1) (var2 base2 exp2) ...)
((test? ...) final-exp)
side-effect-statements ...)
```

Les variables *var1*, *var2*, ... sont créées avec des valeurs initiales *base1*, *base2*, ... Puis le test (*test?* ...) est évalué. S'il est vrai, *final-exp* est retourné. Sinon, *var1*, *var2*, ... sont redéfinies par les nouvelles valeurs *exp1*, *exp2*, ... et on réitère la procédure.