Mobile

Consignes Chaque fonction doit être accompagnée de sa spécification et d'un jeu de test. Les questions impliquent l'écriture de fonctions supplémentaires dont le comportement n'est pas spécifié dans l'énoncé. Elles pourront être des fonctions locales ou globales à votre guise, mais dans les deux cas, elles doivent être aussi accompagnées d'une spécification. Les fonctions locales ne peuvent pas être testées directement, employez les avec prudence.

On peut, bien sûr, utiliser les fonctions définies dans les questions précédentes pour répondre à une question.

Rappel La forme spéciale `verifier` n'est pas décrite dans votre carte de référence. En voici la syntaxe qui complète la section *Grammaire du langage* de la carte de référence.

<forme-spéciale> ::= ...
ou (`verifier` <symbole> <expression-test>*)

Où la grammaire d'<expression-test> est décrite dans la carte de référence, et un <symbole> est n'importe quel nom qui peut être utilisé comme nom de fonction. De telle manière que pour tester la fonction `toto:int→int`, on pourra écrire, par exemple :

```
(verifier toto
    (toto 18)   -> 42
    (toto -7.3) -> ERREUR)
```

Devoir

Dans ce devoir nous considérerons des mobiles. On construit un mobile à un étage en attachant à une barre des objets, représentés par leur masse (en gramme). En négligeant la disposition des objets sur la barre, un mobile à un étage sera représenté par une liste de nombres (une valeur de type `LISTE[Nombre]`). Voici un exemple de mobile à un étage avec trois objets de masse respectives 5, 7 et 8 :

```
(5 7 8)
```

On construit un mobile à deux étages en attachant à une barre des mobiles à un étage. Un mobile à deux étage est donc représenté par une liste de listes de nombres. Voici un exemple de mobile à deux étages constitué de 3 mobiles à un étage :

```
((5 7 8) (8 12) (9 2 9))
```

Finalement, on construit un mobile à trois étages en attachant à une barre des mobiles à deux étages. Un mobile à trois étage est représenté par une valeur de type `LISTE[LISTE[LISTE[Nombre]]]`. Voici un exemple de mobile à trois étages constitué de 3 mobiles à deux étages, eux-même constitués respectivement de 3, 4 et 2 mobiles à un étage :

```
(((5 7 8) (8 12) (9 2 9))
 ((3 6 6) (5 5 5) (12 3) (4 3 4 4))
 ((6 7 8 9) (4 11 11 4)))
```

Dans la suite, un mobile fera référence à un mobile à trois étages. On précisera, par exemple "mobile à deux étages", pour les autres tailles.

Question 1 Écrire une fonction `masse-totale` qui, étant donné un mobile, calcule la masse totale (en gramme) du mobile. On suppose que toute la masse est apportée par les objets attachés au dernier étage du mobile. La masse totale de l'exemple est 180g.

Question 2 Écrire une fonction `tous-masse?` qui, étant donné un mobile à deux étages (c'est-à-dire de type `LISTE[LISTE[Nombre]]`) et une masse m (en gramme), vérifie que tous les mobiles à un étage qui le constitue on exactement la masse m. Par exemple, dans `((5 7 8) (8 12) (9 2 9))`, tous les sous-mobiles pèsent 20g.

Question 3 On dit qu'un mobile à deux étages est équilibré si tous ses sous-mobiles à un étage ont la même masse. On dit qu'un mobile est équilibré si tous ses sous-mobiles à deux étages sont équilibrés *et* ont la même masse. Le mobile de l'exemple est équilibré.

Écrire une fonction `equilibre?` qui, étant donné un mobile (à trois étages), teste si il est équilibré.

Question 4 Écrire une fonction `double-masse` qui, étant donné un mobile, retourne le mobile où la masse de chacun des objets qui en constituent le dernier étage a été doublée. Sur l'exemple, ça donne le résultat suivant :

```
(((10 14 16) (16 24) (18 4 18))
 ((6 12 12) (10 10 10) (24 6) (8 6 8 8))
 ((12 14 16 18) (8 22 22 8)))
```