

Laboratoire 2

Types, Classes de types, Syntaxe

Exercice 2.1 - Types

1. Donnez (de tête) le type des expressions ci-dessous.

1. `3 * pi`

2. `(1.5, "3")`

3. `head "Hello " ++ "World !"`

4. `[[1,2],[]]`

5. `[('a',1),('b',2)]`

2. Donnez une expression pour chacun des types suivants:

1. `[[[Integer]]]`

2. `[(Integer, Char)]`

3. `(Double, [[Integer]])`

4. `((Integer, Integer), [Bool], Double)`

Exercice 2.2 - Motifs

1. Ecrire une fonction d'addition de vecteurs, représentés par une paire (tuple de deux éléments) en utilisant des motifs.
2. Définissez une fonction de produit scalaire de deux vecteurs dans l'espace (triplets).
3. Réécrivez les fonctions `somme` et `produit` sur une liste à l'aide des motifs.
4. Définir les fonctions `head'` et `tail'` à l'aide de motifs.
5. Trouvez le *Cinquième élément* d'une liste à l'aide d'un motif...

Exercice 2.3 - Gardes

1. Définissez la fonction `pgcd` qui calcule le plus grand commun diviseur de deux nombres.
 - On se souvient qu'il suffit de retrancher successivement le plus petit du plus grand jusqu'à ce qu'ils soient identiques pour trouver ce `pgcd`.
2. Ecrivez une fonction qui transforme des secondes en heures, minutes et secondes sans utiliser de division ni modulo.
 - Attention ce problème n'est pas trivial il faut retourner un résultat sous forme de triplet `(h, m, s)` et utiliser un `let (h, m, s)` sur le **reste** pour exprimer le résultat final.

Exercice 2.4 - **where**

1. Définir à l'aide de gardes et d'un **where** qui définit une fonction auxiliaire la fonction `initiales` tel que:

```
initiales "jean" "dupont" → "J. D."
initiales ""      "dupont" → "X. D."
initiales "jean"   ""      → "J. X."
```
2. Inventez une forme géométrique composée de formes élémentaires et définissez une fonction pour calculer sa surface avec un **where** comme pour l'exemple du cylindre donné en cours.
3. Écrivez une fonction qui rend la liste des facteurs premiers de `n`. Si on décompose `n` par des facteurs croissants à partir de 2 ils sont premiers. Par exemple, les facteurs premiers de 60: 60 divisible par **2**, reste 30, 30 divisible par **2**, reste 15, 15 divisible par **3**, reste 5, 5 pas divisible par 4, 5 divisible par **5**, reste 1.
 Résolvez à l'aide d'une fonction auxiliaire `décompose n f` dans une clause **where**.

Exercice 2.5 - case

1. Définissez une fonction `nieme liste n` qui rend une chaîne indiquant le rang et la valeur du `nieme` élément d'une liste:
`nieme [3,5,6] 1 → "3 est le premier de la liste"`
`nieme [3..] 2 → "7 est le second de la liste"`
`nieme [6,5,4] 3 → "4 est le 3eme de la liste"`
2. Réécrivez la fonction `take n liste` en utilisant la forme suivante:
`take' n list = case (n,list) of ...`

Exercice 2.6 - Types

1. Ecrivez puis vérifiez la signature de quelques fonctions que vous avez écrites en solution aux exercices précédents.