

Laboratoire 5

Structures de données

Exercice 5.1 - Personnes

1. Récrire la fonction initiales de l'exercice 2.4 sur le type `Person` tel que défini dans le cours:

```
data Person = Person {  
    prénom :: String,  
    nom    :: String,  
    age    :: Int,  
    taille :: Float  
} deriving (Show)
```

2. Effectuez une recherche dans une liste de personnes à l'aide de **filter**.
3. Vieillissez les personnes d'une liste d'une année avec **map**.
4. Calculez la moyenne d'âge avec rien qu'un pliage...

Exercice 5.2 - Listes

1. Définissez un type `Liste`
2. Définissez les opération d'insertion et suppression sur une liste triée.
3. Dérivez le type `Liste` de `Show` en implémentant une fonction `show` plus lisible que la version automatique de Haskell
Ne cherchez pas nécessairement à obtenir une représentation identique à celle des listes standard de Haskell (`[1, 2, 3]`).
4. Implémentez une fonction de recherche qui retourne la liste des éléments satisfaisant un prédicat.

Exercice 5.3 : Arbres

1. Définissez un type `Arbre`.
2. Définissez l'opération d'insertion d'un élément dans un arbre.
3. Ajoutez une opération de test d'appartenance au type `Arbre`:
`contient :: Arbre t -> t -> Bool`
4. Créez une fonction qui applatit un arbre de type `Arbre` et rend la liste des éléments :
`applatir :: ... => Arbre t -> [t]`