HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Professeur : Nastaran Fatemi
Assitant : Jonathan Bischof

Série d'exercices

Exercice 1

Ecrivez une fonction récursive qui prend une `List[Int]` en paramètre et qui somme tous les éléments de la liste, sans utiliser la fonction définie Math.sum.

Exercice 2

Ecrivez une fonction qui prend un `Vector[String]` et une `List[Int]` en paramètre.
Cette fonction retourne une `Map[Int, String]`, ayant pour clés les éléments de la liste et pour valeurs les éléments du vecteur, en se basant sur la collection la plus petite.

Exercice 3

Définissez un `Vector[Int]` contenant les valeurs de 1 à 1000.
Ecrivez une fonction qui prend un `Int` en paramètre.

- Si cet `Int` est un multiple de trois, retourne son cube.
- Si cet `Int` est un multiple de deux, retourne son carré.
- Si cet `Int` est impair, retourne -1.
- Sinon, le retourne,

Appliquez cette fonction à toutes les valeurs du vecteur.

Exercice 4

Définissez un `Vector[Int]` contenant les valeurs de 1 à 10.
Ecrivez une fonction qui prend un `Int` en paramètre, et retourne le factoriel de ce nombre, en utilisant une implémentation récursive.
Ecrivez une fonction qui prend un `Int` en paramètre, et retourne le nombre de fibonacci correspondant, en utilisant une implémentation récursive.
Appliquez les fonctions au vecteur comme suit :

- Si le nombre est pair, lui appliquer la fonction factoriel.
- Si le nombre est impair, lui appliquer la fonctione fibonacci.

Exercice 5

Définissez un `Vector[Int]` contenant les valeurs de 1 à 5 et une `List[Int]` contenant les valeurs de 1 à 10.
Ecrivez une fonction qui prend le vecteur et la liste en paramètre, et qui retourne un `Vector[List[Int]]`, où chaque liste du vecteur est la liste initiale multipliée par une des composantes du vecteur.

Exercice 6

Voici le code d'une fonction pour trier une liste :

```
def customSort(list: List[Int]): List[Int] = list match {
  case Nil => Nil
  case x :: xs => insert(x, customSort(xs))
}
```

Définissez une `List[Int]` contenant les valeurs (1, 5, 7, 2, 8, 4, 3, 6, 9).
Ecrivez la fonction `insert`, qui permet d'ordonner la liste dans un ordre croissant.
Appelez la fonction `customSort` avec la liste définie.

Exercice 7

Ecrivez une fonction prenant une `List[Int]` et une condition `Int => Boolean` en paramètre.
Cette fonction permet de filtrer la liste passée en paramètre selon la condition définie, en ne conservant que les éléments de la liste validant cette condition.

Vous pouvez vérifier votre implémentation en utilisant la fonction `filter` prédéfinie pour les collections, et vérifier, par exemple, que la liste ne conserve que les nombres impairs.