

Écrit Intermédiaire

Programmation fonctionnelle, Master 2 I2L,
19 octobre 2017

Aucun document autorisé. Durée 45 minutes.

Consignes : Les fonctions sont à écrire dans le langage Haskell.

1 Question de cours

Question 1.a. (3 points) Quelles sont les différences entre un langage de impératif et un langage fonctionnel ?

2 Listes

Les questions de cet exercice sont indépendantes.

Question 2.a. (2 points) Définir une fonction dont les arguments sont un nombre et une liste de nombres et qui calcule le nombre de la liste le plus éloigné du nombre donné. Par exemple, le nombre le plus éloigné de 14 dans la liste `[18, 5, 12]` est 5.

Question 2.b. (2 points) Définir une fonction dont le résultat est une liste qui double les nombres impairs et divise par deux les nombres pairs d'une liste. Par exemple, la liste `[56, 23, 2, 3]` donnera `[23, 46, 1, 6]`.

Question 2.c. (3 points) Définir une fonction qui calcule la différence entre le grand et le plus nombre d'une liste. Par exemple, sur la liste `[56, 23, 2, 3]`, le résultat est 54.

3 Fonction d'ordre supérieure

Les questions de cet exercice ne sont pas indépendantes.

Question 3.a. (3 points) Définir une fonction qui calcule le dernier élément d'une liste.

Question 3.b. (2 points) Ecrire une fonction avec une liste `l` en argument dont le résultat est une fonction qui un argument `x` calcule la différence entre le nombre `x` et le dernier élément de la liste `l`.

Question 3.c. (1 point) Définir une fonction `subder` dont l'argument est une liste et dont le résultat est une liste auquel est soustrait le dernier élément de la liste à tous ses éléments. Par exemple, le résultat de `subder([4, 7, 8, 2])` est `[2, 5, 6, 0]`.

4 Quick sort

Les questions de cet exercice ne sont pas indépendantes.

Question 4.a. (2 points) Définir une fonction `split` qui à partir d'un nombre `x` et d'une liste de nombres crée un couple de listes, la première liste avec les nombres plus petits que `x` et la seconde avec les nombres plus grands que `x`.

Question 4.b. (2 points) Le tri rapide d'une liste consiste à utiliser la tête de la liste comme pivot pour séparer la queue de la liste en deux sous-listes, l'une avec les nombres plus petits que le pivot et l'autre avec les nombres plus grands. Puis ensuite, le tri concatène les sous-listes obtenues dans le bon ordre avec le pivot. Définir la fonction `quicksort` qui réalise le tri rapide d'une liste de nombres.