Programmation fonctionnelle en Haskell - TP 2 : encore des fonctions

Cette semaine, les fonctions seront écrites dans un fichier tp2.hs en utilisant l'éditeur de votre choix. Vous testerez vos fonctions à l'aide de `ghci` en utilisant la commande `:l tp2` pour charger le contenu du fichier dans l'interpréteur. Afin de vérifier que votre code respecte les conventions Haskell, vous utiliserez régulièrement la commande `hlint tp2.hs` dans une console et corrigerez les problèmes détectés.

Exercice 1 : le retour de la factorielle

1. Écrire la fonction factorielle qui prend en paramètre un entier naturel n et qui retourne $n!$.
2. Écrire la fonction récursive exponentielle qui calcule une valeur approchée de e :

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + ...\frac{1}{n!}$$

Exercice 2 : nombre binomiaux

On note par C_n^p le nombre de façons de choisir p objets parmi n avec $0 \leq p \leq n$. C_n^p peut être calculé de 3 manières différentes :

- Par la formule $\frac{n!}{p!(n-p)!}$
- Par une simple récurrence : $C_n^p = C_n^{p-1} \times \frac{n-p+1}{p}$ avec $C_n^0 = 1$
- Par une double récurrence : $C_n^p = C_{n-1}^{p-1} + C_{n-1}^p$ avec $C_n^0 = C_n^n = 1$

Écrire les fonctions `binomiaux1`, `binomiaux2` et `binomiaux3` qui calculent les coefficients binomiaux suivant les trois méthodes.

Exercice 3 : pgcd/ppcm

1. Écrire la fonction récursive `pgcd` qui, étant donné deux entiers naturels a et b, calcule le plus grand diviseur commun de ces deux entiers. Pour calculer le pgcd d'une façon récursive, on utilise l'algorithme d'Euclide :

- $pgcd(a, 0) = a$
- $pgcd(a, b) = pgcd(b, a \bmod b)$ si $b \neq 0$ et $a \geq b$
- $pgcd(a, b) = pgcd(b, a)$ si $a < b$

où $a \bmod b$ désigne le reste de la division euclidienne de a par b.

2. En utilisant `pgcd`, écrire la fonction `ppcm` qui calcule le plus petit multiple commun de deux entiers.

Exercice 4 : puissances

1. Écrire la fonction `puissance` prenant en paramètre deux entiers relatifs x et k et qui retourne x^k. Ne pas oublier le cas où $k < 0$.
2. Écrire la fonction `puissanceDicho` qui fait la même chose que `puissance` mais en utilisant la dichotomie. Cette nouvelle fonction repose sur le fait que si k est pair alors $x^{2n} = (x^n)^2$ et $x^{2n+1} = (x^n)^2 \times x$.

Exercice 5 : opérations sur les vecteurs (2D)

Écrire les fonctions suivantes sur des vecteurs à deux dimensions représentés par des couples :

- produit scalaire $\vec{u}.\vec{v} = u_x * v_x + u_y * v_y$
- norme $\sqrt{\vec{u}.\vec{u}}$
- produit par un réel $\vec{u} * k = (u_x * k, u_y * k)$
- soustraction $\vec{u} - \vec{v} = (u_x - v_x, u_y - v_y)$
- produit vectoriel (renvoyer un vecteur 3D, c'est à dire un triplet) $\vec{u} \wedge \vec{v} = (0, 0, u_x * v_y - u_y * v_x)$

Exercice 6 : les triangles

Écrire une fonction qui détermine la nature d'un triangle (une chaîne de caractère le décrivant). La nature d'un triangle est définie par la relation existant entre les mesures de ses trois côtés. Un triangle est :

- rectangle si ses côtés vérifient le théorème de Pythagore,
- isocèle s'il a deux côtés égaux,
- équilatéral si ses trois côtés sont égaux,
- rectangle isocèle s'il est à la fois rectangle et isocèle,
- Sinon c'est un triangle quelconque.

Donner deux versions (indépendantes) de la fonction :

1. les mesures des trois côtés sont passées en paramètres,
2. les coordonnées (planaires) des trois sommets sont passées en paramètres.

Exercice 7 : les quadrilatères

Écrire une fonction qui détermine la nature d'un quadrilatère donné. Un quadrilatère est délimité par ses quatre sommets.

Un quadrilatère peut être :

- un carré si ses quatre côtés sont égaux et l'un de ses angles est droit
- un losange si ses diagonales sont perpendiculaires,
- un rectangle si ses diagonales sont égales,
- un parallélogramme si ses diagonales se coupent en leur milieu,
- quelconque si aucune des propriétés précédentes n'est satisfaite.