
Programmation fonctionnelle - Interrogation 2

1 avril 2010

Consignes.

Spécifier les fonctions auxiliaires éventuelles, même celles définies localement.

Les fonctions prédéfinies (telles `list`, `append`, `member`, ...) ne doivent pas être spécifiées ni redéfinies.

Répondre à chaque question sur une ou plusieurs feuilles A4 séparées.

Ne pas utiliser de crayon, ne pas utiliser de rouge.

Mentionner nom, prénom, section et numéro de la question sur chaque feuille.

Question 1. Un entier naturel n est *presque parfait* si la somme de tous les diviseurs de n est égale à $2n - 1$.

Par exemple : le nombre 8 est presque parfait car $2 * 8 - 1 = 1 + 2 + 4 + 8 = 15$;

le nombre 12 n'est pas presque parfait parce que $2 * 12 - 1 \neq 1 + 2 + 3 + 4 + 6 + 12 = 28$.

Écrire un prédicat `pperfect?` déterminant si son argument est un nombre presque parfait.

Question 2. Soit f la fonction $\mathbb{N} \rightarrow \mathbb{N}$ définie par :

$$\text{si } n < 4 : f(n) = n^2 + 5$$

$$\text{si } n \geq 4 : f(n) = \{[f(n-1) + f(n-2)] * f(n-4)\} \bmod (1000 + n)$$

Écrire un programme `f` capable de calculer efficacement $f(n)$ à partir de n .

Rappel : Si $m, n \in \mathbb{N}$ et $n \neq 0$, la notation $m \bmod n$ désigne le reste de la division de m par n ; ce nombre est toujours compris entre 0 et $n - 1$.

En *Scheme* on utilisera `(modulo m n)` pour calculer le modulo.

Question 3. Un *arbre n-aire* est un arbre dont chaque nœud admet un nombre quelconque de fils ; tous les nœuds sont étiquetés par des nombres entiers. Une feuille (un nœud qui n'a pas de fils) est représentée par son étiquette ; un nœud interne est représenté par une liste dont le premier élément est l'étiquette du nœud et dont les éléments suivants sont les représentations des fils de ce nœud, de la gauche vers la droite.

On demande d'écrire une fonction `trav-filter` prenant en argument un prédicat unaire `p` et la représentation d'un *arbre n-aire* `t` et qui renvoie la liste des étiquettes rencontrées lors d'un parcours en largeur d'abord et de gauche à droite de `t` pour lesquelles `p` est vrai.

Si `even?` est un prédicat permettant de tester si un nombre entier est pair :

`(trav-filter even? (1 (2 3 (4 5)) (6 (7 8 (9 10))) 11)) ⇒ (2 6 4 8 10)`