

Examen du 22 mai 2015

Veillez lire attentivement les questions. Veuillez rédiger proprement, clairement et de manière concise et rigoureuse.

Question 1 Donner le type des fonctions suivantes :

- 1) `let f1 x y = x :: [y + 1]`
- 2) `let f2 x f = if x then f ()`
- 3) `let f3 g h x = g (h (x + 1)) + h (x + 2)`

Question 2 Écrire la version récursive terminale de la fonction suivante (vous pouvez définir plusieurs fonctions auxiliaires) :

```
let rec f x =  
  if x = 0 then 1  
  else if x < 0 then x * f (x + 1)  
  else x + f (x - 1)
```

Une bibliothèque de livres

Le but de cet exercice est de réaliser un programme pour la gestion d'une bibliothèque de livres.

Chaque livre de la bibliothèque a une fiche. Sur cette fiche, on trouve le titre du livre, son genre (BD, Policier, Roman ou SF) et un numéro d'inventaire unique. Les types OCaml utilisés pour représenter les livres sont les suivants :

```
type genre_livre = Policier | BD | Roman | SF  
type livre = {  
  titre : string;  
  genre : genre_livre;  
  num_inventaire : int  
}
```

Pour chaque emprunt de livre, on mémorise le nom de l'emprunteur, le numéro d'inventaire du livre emprunté et la date d'emprunt. Les types OCaml utilisés pour représenter ces emprunts sont les suivants :

```
type date = { jour : int; mois : int; annee : int }  
type emprunt = { qui : string; quoi : int; date : date }
```

Une bibliothèque est alors simplement constituée de deux listes : celle des livres et celle des emprunts. On utilise le type OCaml suivant pour représenter une bibliothèque :

```
type bibliotheque = {  
  livres : livre list;  
  emprunts : emprunt list; }
```

Question 3 Définir une variable `tintin_1` de type `livre` pour représenter le livre dont le titre est *Tintin en Amérique*, le genre *BD* et le numéro d'inventaire 1.

Question 4 Définir une variable `emprunt_1` de type `emprunt` pour représenter l'emprunt du livre précédent par *Dupont* le 22 mai 2015.

Question 5 Écrire une fonction `string_of_genre`, de type `genre -> string`, qui renvoie le genre d'un livre sous forme d'une chaîne de caractères. Par exemple, `string_of_genre Roman` renverra la chaîne de caractères "Roman". De même, `string_of_genre BD` renverra "BD".

Question 6 En utilisant la fonction précédente, écrire une fonction `affiche_livre`, de type `livre -> unit`, qui affiche la fiche d'un livre à l'écran en indiquant son titre, son genre et son numéro d'inventaire. Par exemple, `affiche_livre tintin_1` affichera :

```
Titre : Tintin en Amérique
Genre : BD
Numéro : 1
```

Question 7 En utilisant la fonction précédente, écrire une fonction `affiche_liste_livres`, de type `livre list -> unit`, qui affiche tous les livres de la liste passée en argument.

Question 8 Écrire une fonction `trouve_livre`, de type `genre -> livre list -> livre list`, telle que `trouve_livre g l` renvoie tous les livres de la liste `l` qui sont du genre `g`.

Question 9 Écrire une fonction `compare_date`, de type `date -> date -> int`, pour comparer deux dates. Ainsi, `compare_date d1 d2` renvoie :

- 1, si `d1` est strictement plus ancienne que la date `d2`
- 0, si les deux dates sont égales
- -1, si `d2` est strictement plus ancienne que la date `d1`

Question 10 En supposant donnée une fonction `ajoute_un_mois`, de type `date -> date`, qui ajoute un mois à une date, écrire une fonction `depasse_date_retour`, de type `date -> date -> bool`, telle que `depasse_date_retour d1 d2` indique si la date `d1` dépasse la date `d2` de plus d'un mois.

Question 11 Écrire une fonction `relance`, de type `date -> bibliotheque -> string list`, qui renvoie la liste des noms des emprunteurs ayant dépassé de plus d'un mois la date pour rendre leur livre. La date passée à la fonction `relance` représentera la date actuelle.