Planche 4

MP2I – l'EISAM

www.crvr.fr

Introduction

Cette planche de colle porte sur le OCaml (v1) et les arbres binaires (v2).

Les exercices sont de difficulté variable et ont pour objectif de vérifier votre compréhension et votre maîtrise de ces notions.

Exercices

Cours

Exercice

VRAI OU FAUX, en justifiant.

- En OCaml, une array se déclare avec `{}`.
- En OCaml, une liste se déclare avec `[[|]]`.
- En OCaml, on concatène des chaînes de caractères avec `s1 + s2`
- En OCaml, l'addition `+` ne marche que sur les entiers.
- En OCaml, le produit `*` marche et sur les entiers, et sur les flottants.

Exercice

- Qu'est-ce qu'un langage compilé ? Interprété ?
- Quelle commande devez-vous taper dans votre terminal Linux pour interpréter du OCaml ?
- Quelle commande devez-vous taper dans votre terminal Linux pour compiler puis exécuter du OCaml ?

Exercice

Rappelez la définition de la signature d'une fonction.

Complétez le code suivant pour que `F` soit de signature `INT -> INT -> INT`.

```
1 let f n =  
2     ...
```

Exercice

```
1 let mystere = add 1;;
```

Que fait ce code :

- Il plante ? Si oui, quelle erreur ? Si non, pourquoi ?
- Si il ne plante pas, que fait `MYSTERE` ?

Exercice

Rappelez les syntaxes pour :

- Déclarer un type `personne` qui a un attribut `age` (`int`) et `nom` (`string`)
- Déclarer un type qui peut soit être `Some n` avec `n` un entier, soit autre chose.

Exercice

Ecrire un code OCaml qui déclare un type arbre qui n'a pas de constructeur de feuille. Comment représenter une feuille avec ?

Exercice

1. Donner la définition d'un arbre binaire.
2. Donner la formule récursive de la hauteur d'un arbre binaire.
3. Encadrer le nombre de noeuds n d'un arbre binaire de hauteur h .

Exercice

Définir arbre binaire de recherche.

Exercice

Ecrire un type OCaml d'arbre.
Déclarer une variable `a` qui est l'arbre vide, `b` qui est une feuille, et `c` qui n'est ni l'un ni l'autre.

Exercice

Définir le parcours postfixe, préfixe et infixé d'un arbre binaire.

Exercices de colle

Exercice

- Après avoir définie les termes en **gras**, répondre à la question suivante :
- Dire quel(s) algorithme(s) sont **totale-ment correctes** et lesquels sont **partiel-lement correctes**.

```
1. let rec contient l x = match l with  
2.   | [a] when a=x-> true  
3.   | e::t -> contient t x;;
```

```
2. let rec contient l x = match l with  
1.   | e::t when e=x-> true
```

```
3 | e::t-> contient t x;;
```

```
3.  
1 let rec contient l x = match l with  
2 | [e] -> e=x  
3 | e::t when e=x-> true  
4 | e::t-> contient t x;;
```

```
4.  
1 let rec contient l x = match l with  
2 | [] -> false  
3 | e::t when e=x-> true  
4 | e::t-> contient t x;;
```

Exercice

Écrivez une fonction `compose` qui prend deux fonctions `f` et `g` et renvoie la composition de ces fonctions. En particulier, la sortie est bien une fonction et non le résultat de l'application de la composition sur une entrée. Donnez sa signature.

Exercice

Écrivez une fonction `map` qui applique une fonction à tous les éléments d'une liste et renvoie la liste résultante. N'utilisez pas la fonction `List.map` de la bibliothèque standard. Donnez sa signature.

Exercice

Écrivez une fonction `filtre` qui prend un prédicat (fonction renvoyant un booléen) et une liste, et renvoie une nouvelle liste contenant uniquement les éléments qui satisfont le prédicat. N'utilisez pas la fonction `List.filter` de la bibliothèque standard. Par exemple le filtre `pair` sur `[1;2;3]` renvoie `[2]`. Donnez sa signature.

Exercice

Écrivez une fonction `est_palindrome` qui vérifie si une liste est un palindrome (se lit de la même façon dans les deux sens). Donnez sa signature.

Exercice

Écrivez une fonction `est_premier` qui vérifie si un entier donné en argument est premier ou non. Donnez sa signature.

Exercice

Écrivez une fonction `fusion` qui prend en entrée deux listes triées et renvoie la liste qui est la fusion triée des deux. Par exemple, `fusion [1;2;5] [3;5]` renvoie `[1;2;3;5;5]`.

Arbres

Exercice

1. Définir un type OCaml `arbre` qui est soit :
 - L'arbre vide
 - Un noeud qui a une valeur (un entier) et deux fils.
2. Ecrire une fonction `est_vide` qui renvoie `true` si l'arbre est vide et `false` sinon.
3. Ecrire une fonction `contient arbre val` qui renvoie `true` si l'arbre contient la valeur, et `false` sinon.

Exercice

1. Définir un type OCaml `arbre` qui est soit :
 - L'arbre vide
 - Un noeud qui a une valeur (un entier) et deux fils.
2. Ecrire une fonction `est_nonvide` qui renvoie `true` si l'arbre n'est pas vide et `false` sinon.
3. Ecrire une fonction `insertion arbre val` qui renvoie l'arbre passé en entrée auquel on a ajouté la valeur donnée (peu importe l'emplacement).

Exercice

Un arbre binaire de recherche est un arbre binaire tel que pour tout noeud $N(x, gauche, droite)$:

- Toutes les valeurs dans l'arbre *gauche* sont plus petites que x
 - Toutes les valeurs dans l'arbre *droite* sont plus grandes (strictement) que x .
1. Définir un type OCaml `arbre` qui est soit :
 - L'arbre vide
 - Un noeud qui a une valeur (un entier) et deux fils.
 2. Ecrire une fonction `est_nonvide` qui renvoie `true` si l'arbre n'est pas vide et `false` sinon.
 3. Ecrire une fonction `recherche arbre val` qui renvoie `true` si l'arbre contient `val` et `false` sinon.
 4. Ecrire une fonction `insertion arbre val` qui renvoie l'arbre passé en entrée auquel on a ajouté la valeur donnée (au bon emplacement).

Exercice

1. Définir un type OCaml arbre qui est soit :
 - L'arbre vide
 - Un noeud qui a une valeur (un entier) et deux fils.
2. Ecrire une fonction `taille` qui renvoie le nombre d'éléments contenu dans un arbre.

Exercice

1. Définir un type OCaml arbre qui est soit :
 - L'arbre vide
 - Un noeud qui a une valeur (un entier) et deux fils.
2. Ecrire une fonction `taille` qui renvoie le nombre d'éléments contenu dans un arbre.

Exercice

Soit le type

```
1 type arbre = Vide | Noeuf of int * arbre * arbre
```

1. Comment représenter une feuille avec ce type ?
2. Ecrire une fonction `hauteur` qui renvoie la hauteur d'un arbre passé en argument.

Barème

Vous n'allez pas assez rapidement en colle, désormais ne traiter que la question de cours ne peut donner que jusqu'à 8/20. Traiter qu'un seul exercice que jusqu'à 12/20. Au-dessus de 12 ça dépendra de votre efficacité, de vos réponses à des questions plus théoriques, et de votre maîtrise du cours.