

UQÀM Université du Québec à Montréal

Paradigme de programmation (inf-2160)

Hiver 2012

Examen intra

7 mars 2012

CONSIGNES

· Les règlements de l'UQAM concernant le plagiat seront strictement appliqués.

· Il est important de bien expliquer vos choix s'il y a lieu.

· La documentation est permise.

· La durée de l'examen est de 3 heures.

· Vous pouvez utiliser les versos comme brouillon ou comme espace supplémentaire.

· Il est interdit de dégrafer le questionnaire.

· Les téléphones cellulaires, calculatrices, ordinateurs, palm, baladeurs, iPods, etc. sont interdits.

#1 _____ / 13

#2 _____ / 12

#3 _____ / 12

#4 _____ / 13

TOTAL

_____ / 50

IDENTIFICATION

NOM : _____

PRÉNOM : _____

CODE PERMANENT : _____

SIGNATURE : _____

GROUPE : _____

PROFESSEUR : _____

commentaire :

Numéro 1. (13 pts)**Objectif(s) :**

- Application des connaissances.
- Fonction récursive.
- Écriture de code Haskell.

Question : Écrivez une fonction qui calcule la somme des valeurs plus grande que 10 dans une liste.
Par exemple :

`somme10 [5, 11, 4, 10, 45] = 11 + 45 = 56`

Numéro 2. (12 pts)**Objectif(s) :**

- Application des connaissances.
- Lecture de code Haskell.
- Application partielle.

Question : Que va afficher le code suivant :

```
a = [1, 3, 5, 7, 9, 11, 13]
```

```
h = ( map g ) . ( map f )
```

```
  where
```

```
    f = (*) 2
```

```
    g = (+) 1
```

```
main = print ( h a )
```

Numéro 3. (12 pts)**Objectif(s) :**

- Application des connaissances.
- Lecture de code Haskell.

Que va afficher le code suivant :

```
f [] = []
```

```
f [x] = [x]
```

```
f (x:y:xs) = x : f xs
```

```
main = print (f [0,5..50])
```

Numéro 4. (13 pts)**Objectif(s) :**

- Application des connaissances.
- Type récursif.
- Écriture de code Haskell.

Soit la déclaration de type suivante :

```
data (Eq a, Ord a) => ArbreBinaire a =  
  Feuille  
  |  
  Noeud  
  {  
    conteneur :: a,  
    gauche    :: ArbreBinaire a,  
    droite    :: ArbreBinaire a  
  }
```

Écrivez le code d'une fonction qui calcule le nombre de `Noeud` dans un `ArbreBinaire`. (Une `Feuille` n'est pas un `Noeud`.)

```
nombreNoeud :: ArbreBinaire a -> Int
```