Nom Prénom

Devoir surveillé

Année 2014/2015

Vous devez répondre dans les cadres prévus à cet effet. Téléphones portables, calculatrices, ordinateurs et tablettes interdits. Notes personnelles (manuscrites ou imprimées) et documents de cours, TD et TP autorisés.

Lambda-Calcul 10 points

Décomposez le terme suivant en encadrant les abstractions en pointillés et les applications en trait continu.

$$\lambda t \ . \ \lambda u \ . \ \lambda q \ . \ t \ q \ (u \ t)$$

10%

Dans le terme suivant, entourez les occurrences libres de variables et indiquez les occurrences de variables liées entre elles.

$$\lambda x \ . \ x \ \lambda x \ . \ x \ y \ \lambda y \ . \ y \ y$$

5%

Parmi les termes suivants, entourez ceux qui sont équivalents entres eux.

$\lambda x.xtx$	$\lambda y.yty$
$\lambda x.xqx$	$\lambda x.xxx$

5%

Réécrivez le terme $\lambda x.x\lambda x.xy\lambda y.xy$ en faisant les renommages nécessaires pour éviter que des occurrences de variables non liée entre elles portent le même nom.

5%

Un renommage est il nécessaire pour réduire le terme $(\lambda y.(\lambda x.xy))(\lambda y.xy)$? Si oui, réalisez ce renommage. Dans tous les cas, donnez la première étape de l'évaluation du terme.

10%

Détaillez l'évaluation complète du terme $(\lambda x.xy)(\lambda x.xy)(\lambda x.xy)$.

15%

CAML 10 points

Vos programmes CAML doivent être en style fonctionnel pur, sans instruction itérative. Il sera tenu compte de la simplicité des solutions proposées.

Soit les fonctions inv et h définies de la manière suivante :

```
let rec inv f x r = if x>=r then 0 else 1 + inv f (f x) r;;
let h x = 3*x;;
```

Donnez le résultat de l'exécution de inv h 2 5 et de inv h 2 17.

10%

Complétez ci-dessous la définition de la fonction **intercl** de telle sorte que si a et b sont des listes d'entiers *classées en ordre croissant* alors intercl a b retourne la liste comportant toutes les valeurs de a et de b classées en ordre croissant. Par exemple intercl [1;3;10;12] [2;11;13;14;21] doit retourner [1;2;3;10;11;12;13;14;21].

```
let rec intercl a b =
match a with
|[]-> 
|ta::qa -> match b with
           |[]-> 
           |tb::qb -> if ta <= tb then
                      else
;;
```

15%

Que réalise la fonction f suivante ?

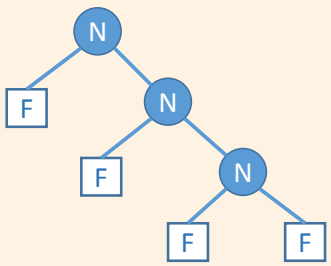
```
let f x w = intercl [x] w;;
```

5%

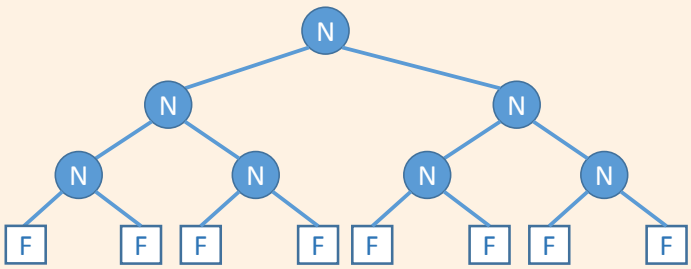
On définit le type arbre de la manière suivante :

```
type arbre = F | N of arbre * arbre;;
```

On dit qu'un arbre est un peigne droit si les fils gauches de tous les noeuds sont des feuilles. On dit qu'il s'agit d'un arbre complet si tous les chemins partant de la racine ont la même profondeur.



Peigne droit



Arbre complet

Définissez la fonction testPeigne telle que si w est un arbre alors testPeigne w retourne vrai si w est un peigne droit et faux sinon.

Définissez la fonction makeCompl telle que si n est un entier alors makeCompl n retourne un arbre complet de profondeur n. Par exemple pour n=0 la valeur retournée est F (une feuille) et pour n=3 c'est la représentation par le type arbre de l'arbre complet donné en exemple ci dessus.

```
let rec testPeigne w =
```

10%

```
let rec makeCompl n =
```

10%