Université Mentouri de Constantine **Le 15/02/2009**
Faculté des Sciences de l'Ingénieur **Section : 4ième Année ing.**
Département d'Informatique **Durée : 1h30**

Sémantique Formelle et Paradigmes des langages de Programmation
Contrôle N°01

***N.B** : Répondre de manière précise et concise.*

Questions de cours (9 pts)

1. Est–il possible qu'un sous programme possède plus d'un point d'entrée ? Si la réponse est positive, par quel mécanisme ? (1 pt)
2. Expliquer avec un exemple la notion de dépendance temporelle dans le paradigme impératif (1pt)
3. Expliquer la stratégie d'évaluation paresseuse (lazy evaluation), donner un exemple (1 pt)
4. En quoi consiste la curryfication d'une fonction ? donner un exemple ? (1 pt)
5. Expliquer la notion de fonction d'ordre supérieur ? donner un exemple en Haskell? (1 pt)
6. Quels sont les avantages et les inconvénients d'une gestion d'exceptions basée sur la reprise ? (1pt)
7. Pourquoi un programme logique procure un niveau d'abstraction plus élevé qu'un programme fonctionnel ou impératif (1 pt)
8. Quelle est la différence entre la durée de vie et la portée d'une variable ?
9. Dans quel cas on peut simplifier algébriquement l'expression suivante ? (1 pt):

x+ f(x,y) +x+y+f(x,y) en 2x+2f(x,y)+y

Exercice N°01 : (3 pts)

Construire la preuve de type des expressions suivantes :

1. **let y = λx →x in y x.**
2. **le type de la fonction twice définie par : twice f x = f f x**

Exercice N°02 (3 pts) :

1) Soit TAB1, TAB2 deux types définis comme suit :

```
Type TAB1 = array [1..10] of integer;
     TAB2 = array [1..10] of integer;
Var a, b : array [1..10] of integer;
     c, d: TAB1;
     e: TAB2
     f: TAB2
```

Quels sont le types compatibles si on applique : a) l'équivalence nominative, b) l'équivalence structurelle, et c) l'équivalence déclarative. (1.50 pt)

2) Ecrire la fonction **" !! "** qui permet de sélectionner le nième élément d'une liste (Exemple: [1,2,3,4,5]**!!**2 = 3) de deux manières :

a) En utilisant la récursivité et le pattern matching, (1 pt)
b) En utilisant la fonction drop, et head (0.50 pt)

Exercice N°03 (5 pts)

Soit la fonction foldr définie par :

```
foldr f  v [ ] = v
foldr f  v [ x : xs] = f x (foldr f v xs )
```

1. Quel est le résultat de l'application: foldr (+) 0 [1,2,3,4], où (+) est l'opérateur de l'addition?(1pt)
(**Aide Input:** foldr (/) 2 [8,12,24,4] Output: 8.0)
2. A quelle fonction correspond foldr (+) 0 ? Définir récursivement cette fonction.
3. A quelle fonction correspond foldr (*) 1 ? Définir récursivement cette fonction
4. Quel est le type de foldr ? (2 pt):

Bonne Réussite