Université BORDEAUX I **Année 1995-96**
Session de septembre

Maîtrise d'informatique

Modèles et Techniques de Programmation (J.P.Braquelaire)

Durée: 2 heures
Tous documents autorisés

Première partie

1) – 3 points – Quels mécanismes de liaison d'un symbole à une définition connaissez-vous ? Donner pour chaque mécanisme deux exemples de définition (en utilisant deux langages de programmation différents), et un exemple d'évaluation.

2) – 3 points – Donner trois stratégies d'évaluation des appels de fonction (le passage de paramètres par référence n'est pas considéré comme une stratégie d'évaluation fondamentalement différente de l'appel par valeur). Donner pour chaque stratégie un exemple en expliquant le résultat obtenu.

3) – 6 points – On considère le programme suivant écrit dans un langage de programmation imaginaire comportant un mécanisme de définition et d'évaluation de fonction

```
DEF f : x --> (x , f(x)) FED
DEF g : --> DEF f : x --> "b" FED ; "a" FED
f(g)
```

La première ligne définit une fonction de nom `f`, avec un paramètre formel `x`, et dont le résultat de l'évaluation est la liste dont le premier élément est `x`, et le second élément est le résultat de l'évaluation récursive `f(x)`. La seconde ligne définit la fonction sans paramètre `g` dont l'évaluation provoque une définition de fonction, puis retourne la chaîne `a`. La troisième ligne demande l'évaluation de l'expression `f(g)`. On peut, selon la stratégie d'évaluation et le mécanisme de liaison retenus, obtenir les valeurs suivantes:

- `("b")`
- `("a", "b")`
- la liste infinie `("a", "a", "a", ...)` (l'évaluation boucle)

Expliquer chacun de ces cas.

T.S.V.P.

Deuxième partie

On souhaite écrire en C standard un module permettant de gérer des listes potentiellement infinies. On utilise pour cela une cellule dont la valeur est soit une valeur générique, soit une référence à une fonction permettant de calculer la valeur suivante.

5) – 3 points – Écrire la structure de données codant une cellule d'une telle liste.

6) – 5 points – Proposer une interface pour ce module (les fonctionnalités sont les fonctionnalités classiques d'une liste : initialisation, passage à l'élément suivant et obtention de la valeur, à ceci prés que la liste est construite par nécessité). On donnera la définition de l'interface et un exemple simple de son utilisation. On ne demande pas d'écrire l'implémentation du module.