

Contrôle d'Informatique

Les Principes des Langages de Programmation (INF 321)

Promotion 2004

Sujet proposé par Gilles Dowek

5 juillet 2005

Les exercices qui suivent sont indépendants et peuvent être traités dans n'importe quel ordre. On attachera une grande importance à la clarté, à la précision et à la concision de la rédaction.

Exercice 1

On définit la classe des listes

```
class List {
    int hd;
    List tl;
    List(int x, List y) {this.hd = x; this.tl = y;}}
```

1. On considère une fonction `compte` qui prend en argument deux listes `l1` et `l2` et retourne un entier qui est le nombre d'éléments de la liste `l1` qui sont également dans `l2` et à la même place, c'est-à-dire le nombre d'entiers i tels que le $i^{\text{ème}}$ élément de la liste `l1` soit égal au $i^{\text{ème}}$ élément de la liste `l2`. Quelle est la valeur de l'expression

```
compte(new List (1, new List (2, new List (3, new List (4,null)))),
        new List (5, new List (2, new List (1, new List (4,null)))))
```

?

2. Écrire la fonction `compte` en Java.

Exercice 2

On définit la classe des listes comme ci-dessus. On exécute les instructions suivantes.

```
List l1 = new List (2,new List(3,new List(4,null)));
List l2 = new List (1,l1.tl);
```

1. Quels sont l'environnement et la mémoire construits ?
 2. Représenter graphiquement cet état.
- On exécute ensuite l'instruction suivante.

```
l2.tl.tl.tl = l1;
```

3. Quels sont l'environnement et la mémoire construits ?
4. Représenter graphiquement cet état.

Exercice 3

On définit la classe des arbres binaires

```
class Arbre {
  Arbre gauche;
  Arbre droit;

  Arbre (Arbre x, Arbre y) {this.gauche = x; this.droit = y;}}
```

Et on définit les arbres

```
Arbre a = new Arbre(new Arbre(null,null), new Arbre(null,null));
Arbre b = new Arbre(new Arbre(new Arbre(null,null),null),
                    new Arbre(null,null));
```

1. Représenter graphiquement ces arbres.
2. On rappelle que la taille d'un arbre est le nombre de ses nœuds et que sa hauteur est la longueur de sa plus longue branche diminuée de 1. Quelle sont la taille et la hauteur de l'arbre a? Et de l'arbre b?
3. Écrire en Java une fonction qui prend en argument un arbre et retourne sa taille. Écrire en Java une fonction qui prend en argument un arbre et retourne sa hauteur.

Exercice 4

On définit la classe des arbres binaires comme ci-dessus. Un arbre binaire est dit *complet* si les sous-arbres gauche et droit de chaque nœud sont de même hauteur.

1. Écrire une fonction qui prend en argument un arbre et retourne un booléen qui indique si l'arbre est complet ou non. La complexité de cette fonction doit être linéaire en le nombre de nœuds de l'arbre.
3. Quel est le nombre de nœuds d'un arbre complet de hauteur h?

Exercice 5

On rappelle qu'en Java, une chaîne de caractères s'écrit entre guillemets : "bonjour" et que la fonction de concaténation sur les chaînes de caractères s'écrit + : "bon" + "jour".

1. Écrire en Java une fonction enum qui prend en argument un entier n et affiche les 2^n chaînes de caractères différentes de longueur n et formées des lettres a et b.

Par exemple, l'instruction `enum(10)` ; affichera les 1024 chaînes de caractères `aaaaaaaaaa`, `aaaaaaaaaab`, `aaaaaaaaaaba`, `aaaaaaaaaabb`, `aaaaaaaaabaa`, ...

Exercice 6

On considère un environnement e, une mémoire m, un environnement global G et une liste C de types construits qui contient au moins un type List qui a deux champs hd de type int et

`tl` de type `List` et au moins le constructeur `List ()` qui construit un enregistrement dont le champ `hd` vaut `0` et le champ `tl` vaut `null`.

1. On évalue l'expression `new List ()` dans e, m, G et C . Quelle est la mémoire à la fin de cette évaluation ?

2. Quelle est la valeur résultat de cette évaluation ?

3. On rappelle que si t est une expression, alors $\Sigma(t, e, m, G, C)$ est un couple formé de la valeur de cette expression dans e, m, G et C et de la mémoire à la fin de l'évaluation de t dans e, m, G et C .

Quel est le couple $\Sigma(\text{new List } (), e, m, G, C)$ dans le cas où C contient la définition du type `List` décrite ci-dessus ?

Exercice 7

1. En Java, qu'affiche le programme

```
class Point {
    int x;
    int y;

    Point (int a, int b) {this.x = a; this.y = b;}
}

class Prog {

    static void proj (Point p) {p.y = 0;}

    public static void main (String [] args) {
        Point p = new Point (1,2);
        proj(p);
        System.out.println(p.y);}

    ?
}
```

2. En C, qu'affiche le programme

```
struct Point {int x; int y;};

void proj (struct Point p) {p.y = 0;}

int main () {
    struct Point p = {1,2};
    proj(p);
    printf("%d\n",p.y);
    return 0;}

?
```

3. En Caml, qu'affiche le programme

```
type point = {mutable x : int; mutable y : int};;
```

```
let proj p = (p.y <- 0)
```

```
in let p = {x = 1; y = 2;}
```

```
in (proj p; print_int p.y; print_newline());;
```

?