

# Composition d'Informatique

## Les Principes des Langages de Programmation (INF 321)

Promotion 2006

Sujet proposé par Gilles Dowek

9 juillet 2007

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 (4 points)

Soit le type des listes d'entiers.

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

1. Quelle est la liste des décimales du nombre  $7/55$ ?
2. Combien cette liste a-t-elle de sous-listes distinctes? Quelles sont-elles?
3. Quel est le nombre minimal de cellules qui permet de représenter cette suite sous la forme d'une liste rationnelle?
4. Écrire un programme en Java qui construit cette liste rationnelle.

### Exercice 2 (4 points)

Soit le type des arbres binaires avec un contenu entier.

```
class Arbre {
    Arbre g;
    int c;
    Arbre d;
    Arbre (Arbre x, int y, Arbre z) {g = x; c = y; d = z;}}
```

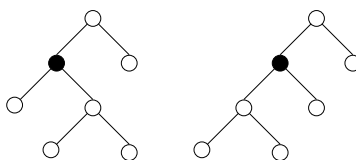
1. Écrire une fonction qui fait le produit du contenu de tous les nœuds d'un arbre.
2. Quelle est la valeur de cette fonction sur l'arbre `new Arbre(new Arbre(null, 5, null), 6, new Arbre(null, 7, null))`?
3. On veut améliorer l'efficacité de cette fonction. Modifier cette fonction de manière à ce qu'elle lève une exception quand l'arbre contient un 0.
4. Écrire une fonction qui a le même comportement que la fonction de la question 1., mais s'interrompt et retourne 0 dès qu'elle rencontre un 0 en parcourant l'arbre.

### Exercice 3 (5 points)

On considère le type des arbres binaires sans contenu.

```
class Arbre {  
    Arbre g;  
    Arbre d;  
    Arbre (Arbre x, Arbre y) {g = x; d = y;}}
```

On dit qu'un arbre est obtenu à partir d'un autre par un échange s'il est obtenu en échangeant les deux sous-arbre d'un nœud quelconque. Par exemple, les deux arbres ci-dessus peuvent être obtenus l'un à partir de l'autre par un échange (au nœud marqué en noir).



On dit que deux arbres sont équivalents si on peut transformer l'un en l'autre par une suite d'échanges.

1. Écrire une fonction qui teste l'équivalence de deux arbres.
2. Donner deux arbres qui ont le même nombre de nœuds mais ne sont pas équivalents.

### Exercice 4 (4 points)

Soit la fonction

```
static void f (int x, int n) {  
    if (x != 0) {if (n == 0) System.out.print("."); else f(10,n-1);  
    f(x-1,n);}}
```

1. On appelle  $u(x, n)$  le nombre de points affiché par l'appel  $f(x, n)$ . Donner une relation de récurrence pour la fonction  $u$ .
2. Montrer que  $u(x, 0) = x$  et que  $u(x, n+1) = x u(10, n)$ .
3. Montrer que  $u(10, n) = 10^{n+1}$ .
4. Quelle est la valeur de  $u(x, n)$  ?
5. Si on écrit mille points par seconde, combien durera l'exécution de  $f(1, 20)$  ?

### Exercice 5 (3 points)

1. Soit le programme en Java

```
class Point {  
    int x;  
    int y;  
    Point(int n, int p) {x = n; y = p;}}
```

```

class Prog {

static void proj (Point p) {p.y = 10;}

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

```

Donner l'environnement  $e$  et la mémoire  $m$  dans lesquels l'instruction `proj(a);` est exécutée.  
 Donner l'environnement  $e'$  et la mémoire  $m'$  dans lesquels l'instruction `p.y = 10;` est exécutée.

Quelle est la valeur de  $\Sigma(\text{proj}(a);, e, m)$  ?

Que le programme affiche-t-il ?

2. Mêmes questions avec le programme C

```

struct Point {int x; int y;};

void proj (struct Point p) {p.y = 10;}

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

```