

# Composition d'Informatique

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

Promotion 2009

Sujet proposé par Gilles Dowek

13 juillet 2010

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 (5 points)

On représente des ensemble finis d'entiers par des listes de booléens : la liste `l` représente l'ensemble des entiers `n` tels que le `nème` élément de la liste `l` soit égal à `true`. On se donne, pour cela, la classe des listes de booléens

```
class List {
  boolean hd;
  List tl;

  List (boolean h, List t) {hd = h; tl = t;}}
```

1. Quel est l'ensemble représenté par la liste `true, true, false, true` ?
2. Et par la liste `true, true, false, true, false` ?
3. Écrire une fonction qui prend en argument deux listes de booléens représentant deux ensembles `A` et `B` et calcule une liste représentant l'intersection de ces deux ensembles.
4. Écrire une fonction qui prend en argument deux listes de booléens représentant deux ensembles `A` et `B` et calcule une liste représentant la réunion de ces deux ensembles.
5. Écrire une fonction qui prend en argument deux listes de booléens représentant deux ensembles `A` et `B` et retourne un booléen indiquant si l'ensemble `A` est un sous-ensemble de l'ensemble `B` ou non.

### Exercice 2 (4 points)

On reprend le type des listes booléens de l'exercice précédent.

1. Que calcule la méthode suivante ?

```
static int longueur (List l) {
  if (l == null) return 0;
  return 1 + longueur(l.tl);}
```

2. Transformer cette méthode en une méthode dynamique.
3. Quelle est l'unique liste à laquelle la méthode statique peut s'appliquer mais pas la méthode dynamique ?

### Exercice 3 (3 points)

On considère la classe des arbres

```
class Arbre {
  int val;
  Arbre gauche ;
  Arbre droite;

  Arbre (Arbre x, int y, Arbre z) {gauche = x; val = y; droite = z;}
}
```

1. Que calcule la méthode suivante ?

```
static int produit (Arbre a) {
  int n, p;
  if (a == null) return 1;
  n = produit(a.gauche);
  p = produit(a.droite);
  return a.val * n * p;}
}
```

2. Comment transformer cette méthode de manière à interrompre le calcul quand un 0 est rencontré ?

### Exercice 4 (6 points)

On considère la fonction

```
static int f (int x) {
  int n = 0;
  while (x <= 100 || n > 0) {
    if (x <= 100) {x = x + 11; n = n + 1;}
    else {x = x - 10; n = n - 1;}
  }
  return x;}
}
```

1. Montrer que quand appelle la fonction  $f$  avec un argument supérieur à 100, la fonction  $f$  retourne ce même argument.

On suppose dans le reste de l'exercice que l'argument de la fonction est inférieur ou égal à 100.

2. Soit  $s = [x = a, n = b]$  un état tel que

- $b \geq 0$
- $a - 10b \leq 101$ .

On suppose, en outre, que la condition d'exécution du corps de la boucle est vérifiée c'est-à-dire que

- $a \leq 100$  ou  $b > 0$ .

On exécute le corps de la boucle

```
if (x <= 100) {x = x + 11; n = n + 1;}
else {x = x - 10; n = n - 1;}
}
```

dans l'état  $s$  ce qui produit un état  $s' = [x = a', n = b']$ . Montrer que

- $b' \geq 0$
- $a' - 10 b' \leq 101$ .

En déduire que dans tous les états  $s = [x = a, n = b]$  produits au cours de l'exécution de la boucle, on a  $b \geq 0$  et  $a - 10 b \leq 101$ .

3. En déduire que dans tous les états  $s = [x = a, n = b]$  produits au cours de l'exécution de la boucle, on a  $2a - 21b \leq 202$ .

4. On exécute le corps de la boucle

```
if (x <= 100) {x = x + 11; n = n + 1;}
else {x = x - 10; n = n - 1;}
```

dans l'état  $s$  ce qui produit un état  $s' = [x = a', n = b']$ .

Montrer que  $2a' - 21b' = (2a - 21 b) + 1$ .

En déduire que la boucle termine.

5. Soit  $s = [x = a, n = b]$  l'état produit par l'exécution de cette boucle. Montrer que  $a = 101$ .

6. Simplifier l'expression de la fonction  $f$ .

## Exercice 5 (2 points)

1. En Java, qu'affiche le programme suivant ?

```
class Sing {
    int x;

    Sing (int a) {x = a;}}

class Prog {
    static void reset (Sing p) {p.x = 0;}

    public static void main (String [] args) {
        Sing p = new Sing (1);
        reset(p);
        System.out.println(p.x);}}
```

2. En C, qu'affiche le programme suivant ?

```
struct Sing {int x;};

void reset (struct Sing p) {p.x = 0;}

int main () {
    struct Sing p = {1};
    reset(p);
    printf("%d\n",p.x);
    return 0;}
```