

## DS2 - Algorithmes et Structures de Données

Mardi 17 Janvier 2017

Durée 3H – Documents non autorisés

### Exercice 1 – Listes chaînées (4pts)

Ecrivez en pseudo-langage une fonction qui calcule le plus grand écart (valeur absolue de la différence de deux éléments) dans une liste chaînée de réels de type `liste-r`, en une seule passe de la liste. **Expliquez.**

```
Type liste-r = ^cellule
    cellule = Enregistrement
                val : réel
                suiv : ^cellule
    FinEnregistrement
```

### Exercice 2 – Pointeurs (8 pts)

On souhaite pouvoir gérer des biens immobiliers, leur propriétaire et leur habitant.

Vous utiliserez les structures de données suivantes :

- `bien` représente le bien immobilier, `numb` est le numéro (clé) du bien, `adrb` est l'adresse du bien, `proprio` et `hab` sont des pointeurs sur les personnes correspondant au propriétaire et à l'habitant du bien.

- `pers` représente une personne, `nomp` est son nom, `adrp` est son adresse et `tbien` est un tableau de pointeurs sur les biens que possède cette personne (elle en possède au plus 10, mais peut ne pas en posséder : tous les pointeurs sont mis à `nil`).

- la liste `lp` est **triée dans l'ordre croissant** sur le champ `nomp`.

```
Type bien = Enregistrement
    numb, adrb : Chaîne
    proprio, hab : ^pers
    FinEnregistrement
lbien : ^bien
pers = Enregistrement
    nomp, adrp : Chaîne
    tbien : tableau [1..10] de lbien
    FinEnregistrement
lpers = ^pers
Var lb : lbien
lp : lpers
```

2.1. Ecrivez en pseudo-langage une procédure qui affiche tous les propriétaires et le nombre de biens qu'ils possèdent.

2.2. Ecrivez en pseudo-langage une procédure qui affiche toutes les personnes qui habitent un bien qu'elles possèdent.

2.3. On souhaite ajouter un **nouveau** bien de numéro `nb` et d'adresse `adrb`, appartenant à une personne de nom `np` et habité par une personne de nom `nh`.

2.3.1. Faire un dessin.

2.3.2. Ecrivez en pseudo-langage une fonction qui renvoie un pointeur sur une personne de nom `n` dans la liste `lp` et qui l'ajoute à `lp` si elle n'existe pas (en demandant son adresse à l'utilisateur et sans biens immobilier), en renvoyant le pointeur.

2.3.3. Ecrivez en pseudo-langage une procédure qui réalise l'ajout d'un nouveau bien. Les personnes peuvent ne pas exister dans la liste `lp`. Dans ce cas, vous devrez les y ajouter (`lp` est triée). Si le propriétaire existe, alors il possède moins de 10 biens.

**Exercice 3 - Arbre Binaire de Recherche (4 pts)**

On souhaite écrire une **procédure récursive** qui affiche dans l'ordre **décroissant** les éléments d'un arbre binaire de recherche.

3.1. Expliquez en français le principe récursif.

3.2. Ecrivez en pseudo-langage cette procédure en utilisant les opérations définies sur le TAD ArbreBinaire.

**Exercice 4 - Dessin récursif d'un arbre binaire (4 pts)**

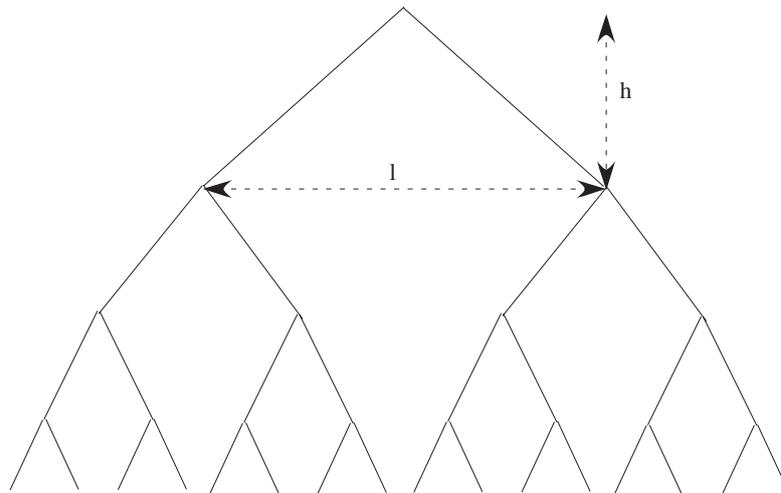
On souhaite dessiner récursivement un arbre binaire parfait (les feuilles sont toutes situées sur le dernier niveau). On dispose des outils suivants :

Procédure va-en(E  $x, y$  : entier) qui permet de déplacer le crayon au point de coordonnées  $(x, y)$ .

Procédure trace-ligne(E  $x, y$  : entier) qui permet de tracer un trait partant du point courant du crayon jusqu'au point de coordonnées  $(x, y)$ .

Pour ce dessin récursif, on suppose qu'à chaque niveau, la distance  $l$  est divisée par 2 et la hauteur  $h$  est multipliée par  $2/3$ .

Exemple :



4.1. Expliquez en français la méthode **récursive** utilisée.

4.2. Ecrivez en pseudo-langage une **procédure récursive** qui réalise ce dessin.