

**Examen final LO44**  
**Durée 2 heures**  
**Documents non autorisés**

**Exercice 1**

1. Ecrire l'algorithme itératif du sous-programme *Apparait*( $l$  : liste,  $x$  : entier,  $n$  : entier) qui teste si un entier  $x$  apparaît au moins  $n$  fois dans une liste d'entiers. Dans le cas particulier où l'entier  $n$  est négatif ou nul, *Apparait* doit donner la valeur Vrai
2. Donner la version récursive du sous-programme *Apparait*.

**Exercice 2**

On considère des listes de points. Rappelons qu'un point est caractérisé par son abscisse  $X$  (un réel) et son ordonnée  $Y$  (un réel).

1. Etant donnée une liste  $l$  de points et deux coefficients réels  $a$  et  $b$ , écrire l'algorithme récursif du sous-programme *Droite* qui construit la liste contenant les points  $(X, Y)$  de  $l$  qui appartiennent à la droite  $Y=aX+b$
2. On considère maintenant des listes de points triés par ordre croissant sur les abscisses. Ecrire l'algorithme du sous-programme *ListeAbscisse* qui à partir d'une liste de points construit une liste où chaque élément est composé d'une abscisse et de la liste des ordonnées des points ayant cette abscisse.  
Exemple : soit la liste des points  $LP = (1,2), (1,6), (3,7), (4,7), (4,6), (4,9)$   
La liste à construire est  $L = (1, (2,6)), (3, (7)), (4, (7,6,9))$
3. Donner en langage C la déclaration complète du type de la liste  $L$  en choisissant une représentation chaînée.
4. Donner la traduction complète en langage C du sous-programme *ListeAbscisse*

**Exercice 3**

Ecrire l'algorithme récursif du sous-programme qui teste si deux arbres binaires ont la même structure (même squelette), c'est-à-dire qu'ils sont identiques si l'on ignore les valeurs des nœuds;