Licence Informatique
Algorithmique Avancée
TD 4: Octobre 2006
Opérations sur des Structures définies récursivement

Dans les 3 premiers exercices il s'agit de structures construites d'enregistrements `liste` de 2 champs `(valeur: entier; queue: ^liste)`; dans les deux premiers, on peut supposer que la liste est terminée par un pointeur `nul`

1. Afficher une liste (une procédure récursive avec comme paramètre un pointeur)
2. Copier une liste (une fonction qui construit une copie de la liste et retourne un pointeur vers la copie.
3. Décider si une liste est circulaire, c'est-à-dire si elle a un pointeur vers un noeud déjà présent;
   (un indice: si deux pointeurs parcourent une liste circulaire, l'un deux fois plus vite que l'autre, le plus rapide va rattraper l'autre quand les deux sont dans la boucle.)

Dans les deux derniers exercices, il s'agit d'enregistrements `noeud` de 4 champs, `(valeur: entier; marqueur: booleen; g,d: ^noeud)`.

4. Copier un arbre (vous pouvez supposer que c'est un vrai arbre sans boucles ni parties partagées)
5. Décider si une structure (pas forcément un arbre) est correctement marquée (c'est-à-dire que tous les champs marqueur des noeuds accessibles de la racine sont `faux`).
   (Oui, on peut le faire en temps linéaire dans la taille de la structure)