Contrôle continu 2007-2008
Pointeurs - Récursivité - Listes

Questions de cours (5 points)
Soit une liste de taille n.

- Quelle est la complexité en temps de la recherche du dernier élément si la liste est simplement chainée ?
- Quelle est la complexité en temps de la recherche du dernier élément si la liste est doublement chainée?
- Dans une liste doublement chainée, pourquoi n'y a-t-il pas de primitive pour supprimer l'élément avant un élément donné ?
- Dessinez la mémoire, après la suite d'opérations suivante :

```
var p :tableau[1..3 ] de ^entier ;
pour i=1 à 3 faire
  new(p[i]) ;
  p[i]^=i ;
finpour
```

Exercice 1 (3 points)
Soit la fonction

```
fonction essai(ref n ; entier, val p :e,tier) ;entier ;
 début
  si n==0  alors
    retourner(p+1)
  sinon
   instruction
   Finsi
 Fin
Finfonction
```

Pour chaque fonction définie par l'instruction ci-dessous dire si la fonction est en récursivité terminale ou non. Expliquez pourquoi.

- retourner(essai(n-1,p*n)+4)
- retourner(essai(n-1,p*n+4))
- retourner(1+essai(n-p,p*n))

Exercice 2 (12 points)
On dit qu'une liste L1 est préfixe d'une liste L2 si la liste L2 commence par la liste L1. Par exemple, la liste L1=(2,4) est préfixe de la liste L2=(2,4,6,8,9).

- Ecrivez une fonction qui teste si L1 est un préfixe de L2.
- Ecrire une fonction qui calcule la liste L2 privée de la liste L1 (dans l'exemple (6,8,9))

Listes simplement chainées (listeSC)

```
fonction premier(val L:type_liste):^type_predefini;
```

fonction suivant(val L:type_liste; val P:^type_predefini):^type_predefini;
fonction listeVide(val L:type_liste):booléen;
fonction créer liste(ref L:type_liste):vide;
fonction insérerAprès(val x:type_prédéfini;ref L:type_liste; val P:^type_predefini):vide;
fonction insérerEnTete(val x:type_prédéfini;ref L:type_liste):vide;
fonction supprimerAprès(ref L:type_liste;val P:^type_predefini):vide;
fonction supprimerEnTete(ref L:type_liste):vide;

Listes doublement chainées (listeDC), On ajoute les primitives suivantes

fonction dernier(val L:type_liste):^type_predefini;
fonction précédent(val L:type_liste; val P:^type_predefini):^type_predefini;