Contrôle continu 2008-2009
Pointeurs - Récursivité - Listes

Questions de cours (4 points)

- Soit une liste de taille n. Quelle est l'occupation en mémoire d'une liste doublement chainée ?
- Quelle est la différence entre le type « pointeur » et le type « curseur » ?
- Dessinez la mémoire, après la suite d'opérations suivante :

```
var p : ^tableau[1..6] de ^entier ;
var c :^entier ;
new(p) ;
pour i=1 à 6 par pas de 2 faire
  new(p^[i]) ;
  p^^[i]=i ;
finpour
```

Dans ce contexte, quel est le problème de la séquence suivante ?

```
var l: entier ;
  l=p^[2]^
```

Exercice 1 (4 points)

Soit la fonction

```
fonction compte(ref T:tableau[1..NMAX] d'entier,val clé :entier) : entier ;
 var p,tmp :entier ;
   début
    tmp=0 ;
    pour p allant de 1 à NMAX faire
          si T[p]<=clé alors
                tmp=tmp+1 ;
          finsi ;
    finpour ;
    retourner(tmp) ;
  fin ;
```

Ecrire cette fonction sous une forme récursive.

Exercice 2 (12 points)

On dit qu'une liste L1 est suffixe d'une liste L2 si la liste L2 finit par la liste L1. Par exemple, la liste L1=(2,4) est suffixe de la liste L2=(6,8,9, 2,4).

- Ecrivez une fonction qui teste si L1 est un suffixe de L2 pour une liste simplement chainée.
- Ecrivez une fonction qui teste si L1 est un suffixe de L2 pour une liste doublement chainée.

Primitives et Fonctions

Liste Simplement Chainées

```
fonction valeur(val L:liste d'objet):objet;
fonction debutListe(ref L:liste d'objet);
fonction suivant(ref L:liste d'objet);
fonction listeVide(val L:liste d'objet): booleen;
```

fonction créerListe(ref L:liste d'objet):vide;
fonction insérerAprès(ref L:liste d'objet; val x:objet;):vide;
fonction insérerEnTete(ref L:liste d'objet,val x:objet):vide;
fonction supprimerAprès(ref L:liste d'objet):vide;
fonction supprimerEnTete(ref L:liste d'objet):vide;
fonction setCléListe(ref L: liste d'objet;val c:curseur):vide;
fonction detruireListe(ref L:lisetd'objet):vide;

Liste Doublement Chainées (ajout de)

fonction finListe(ref L:liste d'objet):vide;
fonction précédent(ref L::liste d'objet): vide;

Fonctions sur les listes

fonction estFinListe(val L:liste d'objet):booléen;
fonction chercher(ref L:liste d'objet; ref x:objet): booleen;
fonction supprimer(ref L:liste d'objet; ref x:objet): vide;