Contrôle continu 2009-2010
Pointeurs - Récursivité – Listes
(pas de documents autorisés)

Questions de cours (6 points)

- Soit une liste de taille n. Quelle est l'occupation en mémoire d'une liste doublement chainée ?
- Ecrire l'algorithme de la primitive insérerAprès dans les listes doublement chainées en implémentation dynamique.
- Dessinez la mémoire, après la suite d'opérations suivante :

```
objet=structure
        un :^entier ;
        deux : listeSC d'entier ;
var p : ^ tableau[1..2] d'objet ;
new(p) ;
pour i=1 à 2 faire
  new(p^[i].un) ;
  p^[i].un^=i+3 ;
  creerListe(p^[i].deux) ;
  insérerEnTete(p^[i].deux, i+1) ;
finpour
```

Dans ce contexte, peut-on écrire

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

Pourquoi ?

Exercice 1 (6 points)

Ecrire une fonction récursive qui calcule a^n pour a réel et n entier positif. On suppose qu'on ne dispose pas de la fonction puissance. Votre fonction est-elle récursive terminale ?

Exercice 2 (8 points)

Ecrire une fonction qui prend en entrée une liste d'entier et qui fournit en sortie une liste telle que tous les nombres multiples de 3 se retrouvent en tête de la liste et les autres éléments sont dans la suite de la liste dans le même ordre que la liste d'entrée. Précisez si la liste de sortie est simplement chainée ou doublement chainée et pourquoi. Donnez la complexité en temps et en mémoire de votre algorithme.

Primitives et Fonctions

Liste Simplement Chainées

fonction valeur(val L:liste d'objet):objet;
fonction debutListe(ref L:liste d'objet):vide;
fonction suivant(ref L:liste d'objet):vide;
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 detruireListe(ref L:lisetd'objet):vide;</pre>

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 appartient(ref L:liste d'objet; ref x:objet): booléen ;