

J1IN3001 ASD1 Devoir surveillé n°1 Durée : 30mn Aucun document autorisé 11/10/2013	Nom : Prénom :
---------------------------------------------------------------------------------------------	-----------------------

Exercice 1 : Du dessin

Soit L une listeSC_car implémentée avec un tableau

L

Taille stock	10
vListe	T
Premier	2
Premier libre	3
Clé	6

T	1	2	3	4	5	6	7	8	9	10
valeur	A	S	B	L	R	U	S	E	F	T
indexSuivant	4	1	5	6	8	10	0	9	0	7

1. Dessinez la liste L correspondante

2. En redessinant les **tableaux et structures** indiquez comment la liste L est modifiée après l'action `supprimerApres(L)`

Exercice 1 : De la lecture

On considère la fonction `mystere` suivante :

```
fonction mystere(ref L:listeSC d'objet):vide;  
  var P:curseur;  
  début  
    P=L.clé;  
    suivant(L);  
    P^.pointeurSuivant=L.cle^.pointeurSuivant;  
    delete(L.cle);  
    L.cle=P;  
  fin  
finfonction
```

1. Que fait la fonction `mystere` dans le cas général ?

Exercice 2 : De l'écriture

On souhaite écrire une fonction qui **supprime dans une liste L toutes les occurrences d'un élément x donné**.

1. Pour implémenter la liste on a le choix entre une liste simplement chaînée d'objet ou une liste doublement chaînée. Dites lequel de ces 2 types abstraits permet d'écrire la fonction souhaitée de façon efficace. Justifiez votre réponse.

2. Décrivez quels sont le(s) cas particulier(s) dont il faudra tenir compte.

3. Ecrivez la fonction `supprimerToutesOccurences` en utilisant les primitives du type abstrait que vous avez choisi à la question 1 et en tenant compte des cas particuliers décrits au point 2.

Liste des primitives du type abstrait liste simplement chaînée.

listeSC= liste de type_predefini;
défini en cours avec les primitives suivantes:

Accès

```
fonction valeur(ref L:listeSC d'objet) : objet;  
fonction debutListe(ref L:listeSC d'objet) :vide;  
fonction suivant(ref L:listeSC d'objet) : vide;  
fonction listeVide(ref L:listeSC d'objet) : booleen;  
fonction estFinListe(ref L:listeSC d'objet) : booleen;
```

Modification

```
fonction creerListe(ref L:listeSC d'objet) : vide;  
fonction insererAprès(ref L:listeSC d'objet, val x:objet;) : vide;  
fonction insererEnTete(ref L:listeSC d'objet, val x:objet) : vide;  
fonction supprimerAprès(ref L:listeSC d'objet) : vide;  
fonction supprimerEnTete(ref L:listeSC d'objet) : vide;  
fonction detruireListe(ref L:listeSC d'objet) : vide;
```

Primitives supplémentaires pour le type abstrait liste doublement chaînée.

```
fonction finListe(ref L:listeDC_car): vide;  
fonction precedent(ref L:listeDC_car) : vide;
```