

<p>Algo 1 Devoir surveillé n°1 Aucun document autorisé 16/10/2007</p>

1. En utilisant tout à tour les deux types d'implémentation vues en cours pour les listes simplement chaînées et **sans écrire** les fonctions, indiquez en justifiant votre réponse, quelle serait la complexité de la fonction `viderListe`
2. Soit une liste de caractères stockés dans une liste simplement chaînée définie par une implémentation dynamique :

```
Cell = structure
    Valeur : car
    Suiv : ^cell
Finstructure
```

```
ListeSC_car = ^cell
```

Soit la liste des primitives associées à ce type abstrait :

```
fonction premier(val L: listeSC_car): ^cell;
fonction suivant(val L: listeSC_car): ^cell;
fonction listeVide(val L: listeSC_car): boolean;
fonction creerListe(ref L: listeSC_car): vide;
fonction insererAprès(val X: car, ref L: listeSC_car, val P: ^cell):
vide;
fonction insererEnTete(val X: car, ref L: listeSC_car): vide;
fonction supprimerAprès(ref L: listeSC_car, val P: ^cell): vide;
fonction supprimerEnTete(ref L: listeSC_car): vide;
```

- 2.1. Rappeler la définition d'une primitive associée à un type abstrait.
- 2.2. Ecrire la primitive : `insererEnTete`
- 2.3. Ecrire une fonction qui supprime dans L toutes les occurrences d'un élément x donné.
Quelle est la complexité de votre fonction ?

3. Soit la fonction mystere suivante :

```
fonction mystere (ref L :ListeSC_car, val x : car) :entier ;
var cpt : entier ;
  p : ^cell ;
début
  si (non liste Vide(L)) alors
    cpt = 0 ; p = L
    tant que (p) faire
      si (contenu(p) = x) alors
        cpt = cpt+1
      finsi
      p = suivant(p)
    fintantque
  finsi
  retourner (cpt)
fin
FinFonction
```

3.1. Que fait la fonction mystere dans le cas général ?

3.2. Donner une version récursive terminale de la fonction mystere.