Travaux Dirigés sur le langage C n°.6

Cours d'Informatique de Deuxième Année

—Deug SSM M3—

Nous abordons dans ce TD la notion de liste chaînée **triée**. Rajoutant aux listes chaînées une structure d'ordre, nous allons pouvoir mettre en évidence ce que peut nous apporter le type "liste chaînée" par rapport par exemple au type "tableau". On s'efforcera ainsi de donner à chaque fois les complexités des algorithmes et on pourra par exemple les comparer avec les complexités des mêmes algorithmes réalisés cette fois sur le type tableau.

Listes chaînées triées...

On considère toujours les déclarations suivantes :

```
typedef struct Cellule
{ int Val;                              # donnee stockee :  un entier.
  struct Cellule  * PtrSuiv;            # pointeur sur la cellule suivante.
} StrCellule;                           # definition du nouveau type :  StrCellule.

typedef StrCellule * PtrCellule;        # definition du nouveau type :  PtrCellule.
```

mais on met une nouvelle sémantique sur ces listes : elles sont désormais interprétées comme des listes chaînées triées.

▸ **Exercice 1**. La notion d'insertion dans une liste triée se caractérise par la propriété suivante : si `lt` est une liste triée et `x` un élément, alors le résultat de l'insertion de `x` dans la liste `lt` est encore une liste triée. On demande dans cet exercice d'écrire une fonction d'insertion de nom `InsertionTriee` réalisant l'insertion d'un élément `x` dans une liste triée `lt`[(1)].

▸ **Exercice 2**. Maintenant que l'on sait ajouter un élément dans une liste triée, on considère l'opération inverse. On demande donc d'écrire une fonction de nom `SuppressionTriee` permettant d'enlever un élément `x` d'une liste triée `lt`.

▸ **Exercice 3**. On cherche dans cet exercice à réaliser la fusion de deux listes triées `lt1` et `lt2`. Écrire une fonction de nom `FusionTriee` réalisant cette opération sans création de nouvelles cellules. Les chaînages des listes initiales seront éventuellement détruits.

▸ **Exercice 4**. Déduire des exercices précédents la fonction réalisant l'algorithme de **tri fusion** sur listes chaînées. Calculer la complexité en temps —voire en espace— d'une telle fonction.

---

(1) On supposera données les fonctions usuelles sur les listes (création d'une nouvelle cellule, ...)