

1. Structures de données	2. Les Listes chaînées	3. Les Files	4. Les Piles	5. Les Arbres	6. Les graphes
--------------------------	------------------------	--------------	--------------	---------------	----------------

### Liste chaînée : Structure – Mémoire – Pointeurs – Fonctions

#### Exercice 1 :

Soit le code source suivant qui définit une structure de données pour traiter les listes chaînées

```
typedef struct Element{
    int contenu;
    struct Element *suivant;
}Element;
```

Soit les programmes suivants :

<pre>void afficherListe(Element *L){     Element *ptr;     ptr=L;     for(ptr=L;ptr!=NULL;ptr=ptr-&gt;suivant){         printf("(%d - %d) - ",ptr,ptr-&gt;contenu);     } }  Element* trouverAdresseFin(){ }  void createElement(Element *L,int c){ }  void insertElement (Element *L,int c){     Element *nouv;     createElement(nouv,c);     if(L==NULL){         L=nouv;     } }</pre>	<pre>int main() {     Element *L;     int taille;      insertElement(L,3);     insertElement(L,7);     insertElement(L,9);     insertElement(L,11);      afficherListe(L);      return 0; }</pre>
--	---

Questions :

1. Donner le résultat du code ci-dessus
2. Sans modifier la structure de données et le type de retour des fonctions, compléter le code ci-dessous pour faire effectivement l'insertion à la fin de la liste à chaque fois
3. Écrit une fonction pour comptabiliser le nombre d'élément d'une liste chaînée
4. Écrit une fonction qui vérifie si une liste chaînée est vide
5. Écrit une fonction pour vider la liste chaînée et libérer l'espace mémoire

**Exercice 2 :**

1. Écrit une fonction qui permet de faire la concaténation de deux chaînes listées chaînées
2. Écrit une fonction qui comptabilise le nombre d'occurrence d'un enregistrement (contenu)
3. Écrit une fonction qui vérifie si un enregistrement existe dans la liste

**Exercice 3 :**

1. Écrit une fonction qui supprime un enregistrement de la liste chaînée (le premier trouvé)
2. Écrit une fonction qui supprime toutes les occurrences d'un enregistrement dans la liste chaînée