TD 7- Structures chaînées

Exercice 1 Soit le type `liste` vu en cours :

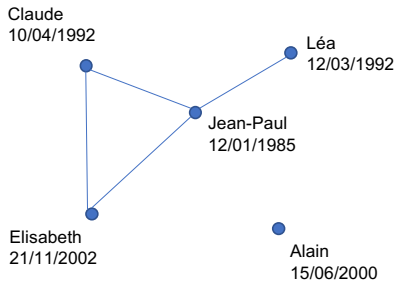
```c
struct cellule {
    int valeur;
    struct cellule *suivant;
};

typedef struct cellule * liste;
```

Écrire les fonctions suivantes :

- Modification de la valeur du n-ième élément d'une liste ;
- Suppression du n-ième élément d'une liste ;
- Concaténation de deux listes (la seconde étant concaténée à la première) ;
- Dupliquer une liste, c'est-à-dire créer une copie réelle de son contenu ;

Exercice 2 Un graphe est une structure mathématique composée d'objets dans laquelle certaines paires d'objets sont en relation. Les objets sont appelés sommets, et les relations entre sommets sont appelées des arêtes. Les sommets sont souvent valués (des informations leur sont associées). Un graphe est fréquemment représenté par un diagramme sous la forme d'un ensemble de points pour les sommets, joints entre eux par des lignes pour les arêtes. Les valuations sont représentées par des étiquettes associées au sommet.
Un réseau social est un graphe dont les sommets sont les personnes et les arêtes sont les liens d'amitié.



Différentes représentations machines sont possibles pour les graphes. Parmi elles, la liste d'adjacence consiste à associer à chaque sommet la liste de ses sommets adjacents ; les différentes étiquettes d'un sommet étant rassemblées dans une structure.

1. Proposer une structure chaînée permettant de représenter un tel graphe par listes d'adjacence. On souhaite de plus avoir un accès à tous les éléments du graphe.
2. Schématiser le résultat de l'allocation mémoire permettant de représenter le graphe ci-dessus dans vos structures. Calculer le nombre d'octets occupés par votre représentation.
3. Écrire une fonction qui étant donné un graphe affiche la liste de ses sommets et pour chacun d'eux son nombre de voisins. Donner la complexité de votre fonction.