

TP N° : 3

Exercice1 :

On souhaite représenter les informations des étudiants d'une classe sous forme d'une Liste Simplyment Chainée (LSC). Chaque *nœud* de la liste représente un étudiant. Un étudiant est représenté par trois champs : *nom*, *prénom* (chaines de caractères : **char***) et une *note* (un réel : **float**).

Pour avoir le contrôle de la liste, on sauvegarde les deux éléments suivants : *Debut* et *Nbr_etudiants* qui représentent respectivement un pointeur vers le début de la liste et une variable entière représentant le nombre d'étudiants de la classe.

- 1) Donner les structures de données à utiliser pour implémenter la liste simplement chaînée et les variables globales à déclarer.
- 2) Ecrire une fonction qui permet d'initialiser la liste chaînée.
- 3) Ecrire une fonction qui permet d'allouer la mémoire d'un nouveau nœud et place les informations d'un étudiant à l'intérieur.
La fonction retourne l'**adresse** de ce nouveau nœud en cas de succès et **NULL** en cas d'échec.
- 4) Ecrire une fonction qui permet d'insérer un nœud au début de la liste.
- 5) Ecrire une fonction qui permet d'insérer un nœud à la fin de la liste.
- 6) Ecrire une fonction qui libère toute la mémoire utilisée par la liste chaînée.
- 7) Ecrire une fonction qui permet de diviser la classe en deux groupes, le 1^{er} groupe contiendra les étudiants admis (ceux qui ont une note ≥ 10) et le deuxième contiendra les non admis (note < 10).
- 8) Ecrire une fonction qui permet d'afficher les étudiants (nom, prénom et note) de chaque groupe (version itérative).
- 9) Ecrire une fonction qui permet de trier les étudiants admis par ordre de mérite.
- 10) Ecrire la fonction main pour tester les différentes fonctions.