TP1 : Listes

SD : Structure de données

L'objectif des 3 premiers TP sera de produire un programme permettant d'identifier les erreurs d'orthographe dans un texte, et ceci par table de hachage. La première étape sera de produire différentes implantations de List et d'en faire la comparaison dans leurs utilisations.

★ Exercice 1: List-O-Choix

▷ **Question 1:** Dans un header nommé **listSimpleChain.h** :

- Définissez un type **ELEMENT** comme étant un pointeur vers un caractère.
- Définissez une structure **cellule** contenant un **ELEMENT val** et un pointeur vers la **cellule** suivante, nommé **suiv**.
- Définissez un type **Liste** comme étant un pointeur de **cellule**.

▷ **Question 2:** Dans le fichier **listSimpleChain.c** correspondant, implantez les fonctions suivantes :

- Création d'une liste vide - **Liste list_creer(void)**
- Vérification si la liste est vide - **int list_estVide(Liste L)**
- Ajout d'un élément en tête de liste - **Liste list_ajoutTete(Liste L, ELEMENT e)**
- Ajout d'un élément en fin de liste - **void list_ajoutFin(Liste L, ELEMENT e)**
- Ajout d'un élément à la nième position de la liste - **void list_ajoutN(Liste L, ELEMENT e, int n)**
- Recherche, dans les n premiers éléments, l'élément de la Liste qui correspond à la chaine de caractère passée en paramètre - **ELEMENT list_rechN(Liste L, char* s, int n)**
- Recherche l'élément de la Liste qui correspond à la chaine de caractère passée en paramètre - **ELEMENT list_rech(Liste L, char* s)**
- Enlève la première occurrence d'un élément d'une liste - **Liste list_remove(Liste L, ELEMENT e)**
- Enlève l'élément à la position n d'une liste - **Liste list_removeN(Liste L,int n)**
- Afficher un élément - **void list_affiche(ELEMENT e)**
- Afficher une liste - **void list_visualiser(Liste L)**

Vous intégrerez les prototypes des fonctions dans le header correspondant.

▷ **Question 3:** Écrivez les fonctions de test permettant de valider cette implantation.

★ Exercice 2: List-1-Choix

▷ **Question 1:** De la même façon que précédemment, vous produirez les fichiers **listContCirc.h** et **listContCirc.c** correspondant à une implantation des listes sous la forme d'un tableau avec gestion circulaire de la liste. Vous apporterez un soin particulier à la gestion des indices de rang dans le tableau ainsi qu'au possible dépassement de taille du tableau (à réaffecter si nécessaire donc...)