

L'objectif de ce TD est de construire un vérificateur orthographique. Dans cette optique, vous devrez tout d'abord créer une structure de liste chaînée avec les fonctions s'y rapportant (Il est possible que nous ayons déjà demandé cela auparavant...). Ensuite, il faudra créer une structure de table de hachage (qui contiendra le dictionnaire à disposition) et les fonctions s'y rapportant. Le dernier élément de ce vérificateur sera la fonction **main** (fournie) permettant de parcourir un texte et de vérifier si chaque mot le composant est présent, ou non, dans le dictionnaire.

★ Exercice 1: List-O-Choix

▷ **Question 1:** A partir du header nommé **l.h**, vous veillerez à créer et compléter un fichier **l.c** contenant les fonctions correspondantes.

Rappel des fonctions :

- Création d'une liste vide - **Liste list_creer(void)**
- Destruction d'une liste - **void list_detruire(Liste l)**
- 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)**
- Recherche, dans les n premiers éléments de L, l'élément qui correspond à la chaîne de caractère passée en paramètre et retourne 0 si l'élément n'est pas trouvé et 1 sinon - **int list_rechN(Liste l, ELEMENT e, int n)**
- Recherche l'élément de la Liste qui correspond à la chaîne de caractère passée en paramètre et retourne 0 si l'élément n'est pas trouvé et 1 sinon - **int list_rech(Liste l, ELEMENT e)**
- Afficher une liste - **void list_visualiser(Liste L)**
- Afficher un élément - **void list_affiche(ELEMENT e)**

★ Exercice 2: Table de Hash, chiche ?

▷ **Question 1:** A partir du header nommé **table.h.h**, vous complèterez le fichier **table.h.c** (fourni) contenant les fonctions correspondantes. Attention, la fonction de hash est fournie.

Rappel des fonctions :

- Création d'une table de hachage vide dont la table d'index est de taille **taille** - **table_h table_creer(int taille)**
- Destruction d'une table de hachage - **void table_detruire(table_h h)**
- Permet de charger les mots du dictionnaire dans la table de hachage h - **void table_charge_dico(table_h h, char* filename)**
- Permet de chercher un mot dans le dictionnaire, renvoie 0 si le mot n'est pas trouvé et 1 sinon - **int table_recherche_dico(table_h h, ELEMENT e)**

★ Exercice 3: Evaluation de performances

▷ **Question 1:** Vous évalueriez les performances de remplissage de la table de hachage en fonction du nombre de sous-table choisi au départ. Vous pouvez étudier le temps d'exécution de la fonction de remplissage en fonction de la taille choisie.

▷ **Question 2:** Vous évalueriez les performances de la fonction de hachage fournie en exemple. Pour ce faire, vous étudierez la distribution des éléments dans la table de hachage. Pensez-vous que la fonction est bien choisie ? Pouvez-vous proposer une autre fonction de hachage permettant de corriger les défauts s'il y a lieu ?