

<p style="text-align: center;"><b>ISIMA 1<sup>ère</sup> ANNEE</b> <b>TP4 de Structure de données – langage C</b></p>
--

Durée : 2 séances - Dossier à rendre pour le 9/06/2017.

**Réalisation d'un gestionnaire d'application multilingue.**

On dispose de fichiers textes contenant chacun des mots dans une langue et son équivalent dans une autre langue (les mots sont séparés par un ;). Exemple de dictionnaire Anglais/Français et Anglais/Espagnol :

**anglais.txt**

hello;bonjour  
bye;au revoir  
sorry;pardon

**espagnol.txt**

hello;ola  
bye;adios  
sorry;disculpe

A partir d'un fichier texte on désire construire un dictionnaire sous la forme d'une table de hachage **indirect**. La table majeure est de taille 29 et la fonction de hachage appliquée sur une clé chaîne de caractères est la suivante :

```
unsigned int hash_string(const char *str)
{
    unsigned int hash = 5381;          /* fonction de hachage de D.J. Bernstein*/
    const char *s;
    for (s = str; *s; s++)
        hash = ((hash << 5) + hash) + tolower(*s);
    return (hash & 0x7FFFFFFF)%HASH_ MAX;
}
```

Avec HASH\_ MAX = 29

Les sous tables sont **des listes chaînées**, chaque cellule donnant un mot, sa traduction et un pointeur suivant.

- 1) Écrire les programmes de gestion de la table de hachage : création de la table, recherche d'une entrée. Ces programmes sont utilisés pour charger un dictionnaire à partir d'un fichier texte.
- 2) Écrire un programme de traduction d'un ensemble de mots (ou d'expressions) dans un langage cible.

bonus : ajouter dans la table majeure un compteur du nombre d'éléments dans la sous-table et fournir en fin de programme la longueur moyenne des sous-tables.

**TRAVAIL A FAIRE :**

1<sup>ère</sup> séance : question 1.

2<sup>ème</sup> séance : question 2.

**REDIGER LE DOSSIER DE PROGRAMMATION CORRESPONDANT :**

**Présentation générale**

- une description de l'objet du TP (3 lignes),
- une description et un schéma de la structure de données et des fichiers de données utilisés,
- l'organisation du code source (dans quel fichier source sont les fonctions).

**Détail de chaque fonction (programme principal compris)**

- les algorithmes de principe et le lexique des notations (paramètres et variables locales en entête)
- les programmes commentés (respecter le guide de style),

**Compte rendu d'exécution**

- **le make file**
- un plan de test et des jeux d'essai complets (données en entrée et résultats obtenus par copie d'écran des structures avec DDD), **tous les cas particuliers seront testés.**

<p>Le TP noté sur le contenu du dossier, la <b>qualité et la modularité du code</b> et la <b>complétude</b> des jeux d'essais.</p>
--