

TD4 : Langage C (3 et fin)

Exercice 1

Ecrire un programme qui lit une chaîne de caractères sur l'entrée standard (avec **gets**), la recopie (avec **strcpy**) dans une autre chaîne, puis affiche la nouvelle chaîne (avec **puts**) et sa longueur (avec **strlen** et **printf**).

Exercice 2

Ecrire un programme qui lit deux chaînes de caractères sur l'entrée standard, puis cherche si la deuxième chaîne fait partie de la première, et affiche un message en conséquence. Vous n'utiliserez pas de fonctions de bibliothèques autres que **gets** et **puts**.

Exercice 3

Reprendre l'énoncé de l'exercice précédent, mais cette fois vous utiliserez les fonctions de bibliothèque **strlen** et **strncmp**.

Exercice 4

On désire connaître le nombre d'occurrences de chaque mot lu sur l'entrée standard. Par exemple, le texte *"il fait beau, n'est-il pas ?"* est constitué de :

il	: 2 fois
fait	: 2 fois
beau	: 1 fois
n	: 1 fois
est	: 1 fois
pas	: 1 fois

Pour cela, on définit les fonctions :

- **void initialiser_mots (void)**
qui initialise la structure de données associée aux mots ;
- **struct mot *trouver_mot (char texte[])**
qui trouve un mot ou le crée s'il n'est pas trouvé ;
- **void mettre_a_jour_mot (struct mot *mot)**
qui met à jour (incrmente) le compteur associé au mot ;
- **void afficher_mots (void)**
qui affiche le résultat final.

1. Ecrire le programme en supposant que les fonctions ci-dessus existent déjà et sont dans un autre module (fichier). Vous placerez dans un fichier **mot.h** les déclarations des prototypes de ces fonctions. Est-il nécessaire de placer la définition de la structure dans ce fichier ?

2. Ecrire les fonctions ci-dessus en prenant comme structure de données une liste simplement chaînée :

```
struct mot
{
    char *texte ;           /* texte associé au mot */
    int nb_occurrences ;    /* nombre d'occurrences du mot */
    struct mot *suivant ;    /* liste chaînée */
};
struct mot *tete;
```