

## Programmation en C

### TP N°5

#### Exercice 1 : (Longueur d'une chaîne)

Écrire une fonction *strlen* qui calcule la longueur d'une chaîne.

#### Exercice 2 : (Comparaison de chaînes)

Écrire une fonction *strcmp* qui compare deux chaînes de caractères pour l'ordre lexicographique. Cette fonction rendra -1, 0 ou 1 comme le fait la fonction *strcmp* de la librairie standard.

#### Exercice 3 : (Suppressions des espaces)

Écrire une fonction *del-space* qui supprime les espaces et tabulations d'une chaîne de caractères. Cette fonction ne doit pas utiliser de tableau intermédiaire. Le morceau de code

```
char s[] = "bonjour à tous";
del-space(s);
printf("s = \"%s\\n\"", s);
```

doit afficher le résultat `s = "bonjouràtous"`.

#### Exercice 4 :

Définir la structure permettant de représenter un nombre complexe. Définir les fonctions complexes suivantes (passages par valeur) :

*void affiche(Complexe z)*, *double imag(Complexe z)*, *double real (Complexe z)*, *Complexe mul(Complexe z1, Complexe z2)*, *double abs(Complexe z)*, *Complexe saisie()*, *void affiche(Complexe z)*

Réécrire ces fonctions en utilisant le passage par adresses.

Tester les fonctions.

#### Exercice 5 :

Soient le modèle de structure *etudiant* déclarés ainsi :

```
struct etudiant{
    char nom[15];
    char prenom[15];
    int CNE;
    float notes[4];
    float moyenne;
};
```

Écrire un programme qui :

1. Déclare un tableau d'étudiants T.
2. Lit les informations de 4 étudiants.
3. Cherche l'étudiant ayant la plus grande moyenne, puis affiche ses informations.
4. Trie le tableau de structures T en ordre décroissant selon la moyenne.

#### Exercice 6 :

Créer une structure `point{int num; float x; float y; }`

Saisir 4 points, les ranger dans un tableau puis les afficher. Reprendre l'exercice en créant une fonction *saisie* pour saisir un point, et une fonction *affiche* pour d'afficher un point.