Travaux Pratiques d'algorithmique n°1

Cours d'Informatique de Deuxième Année

—Deug SSM M3-MI—

Tableaux et pointeurs

L'objet de cette séance de travaux pratiques est la manipulation des tableaux et des pointeurs. Pour cela, on travaillera le traitement des chaînes de caractères et plus précisément :

- saisie d'une chaîne de caractères ;
- opérations basiques (longueur, copie) ;
- manipulations plus complexes (concaténation, comparaison, ...).

Deux notions nouvelles du langage C seront introduites :

- la définition de types ;
- l'allocation dynamique d'espace mémoire.

Toutes les fonctions programmées devront être commentées.

▸ **Exercice 1.** *Concrêtement, toute chaîne de caractères sera représentée par un tableau de caractères. Pour cela, on définit le type* `Chaine` *de la façon suivante :*

```
typedef char *Chaine;
```

Ecrire une fonction qui permet la saisie au clavier d'une chaîne de caractères (le prototype de la fonction sera `Chaine SaisieChaine(void)`*). L'allocation dynamique de la mémoire nécessaire sera effectuée par la fonction* `malloc`*.*

▸ **Exercice 2.** *Ecrire une fonction prenant comme argument une chaîne de caractères et qui affiche celle-ci à l'écran.*

▸ **Exercice 3.** *Ecrire une fonction qui concatène deux chaînes de caractères. Le prototype de la fonction sera* `Chaine Concat(Chaine ch1, Chaine ch2)`*. (Il pourra être utile de programmer une fonction qui calcule la longueur d'une chaîne).*

▸ **Exercice 4.** *Ecrire une fonction qui supprime tous les caractères blancs d'une chaîne de caractères. L'argument de la fonction sera la chaîne initiale, et le retour de la fonction sera la chaine obtenue :* `Chaine SuppressionEspaces(Chaine ch1)`*.*

▸ **Exercice 5.** *Ecrire une fonction qui compare deux chaînes de caractères (pour l'ordre lexicographique). Par exemple : tata $<$ toto et zoo $<$ zoom. Le prototype de la fonction sera* `int Comparaison(Chaine ch1, Chaine ch2)`.
L'entier retourné sera égal à :

- *-1 si ch1 $<$ ch2 ;*
- *0 si ch1 $=$ ch2 ;*
- *1 si ch1 $>$ ch2.*

▸ **Exercice 6.** *Même exercice sans tenir compte des minuscules/majuscules.*

▸ **Exercice 7.** *Ecrire une fonction qui recherche un motif dans une chaîne de caractères. Elle retournera la position du début du motif (si le motif est absent, la fonction devra retourner -1).*