

# TP 5 - Manipulation des chaînes de caractère

## L2 SPI - UE Langage et programmation

Université d'Evry Val d'Essonne

---

### Exercise 1 :

1. Etudier le listing donné en annexe. Que fait-il ? Exécuter le code pour vous aider à comprendre.
2. Exécuter le listing tel quel puis en ajoutant un affichage de l'entier C au fur et à mesure de la saisie. A quoi correspond le nombre affiché ?
3. Calculer la longueur de la chaîne de caractères. Afficher cette longueur à la fin de la saisie.

### Exercise 2 :

1. Saisir, caractère par caractère, une chaîne composée exclusivement de chiffres et afficher la chaîne saisie. Le programme vérifiera que chaque caractère entré au clavier est bien un chiffre compris entre 0 et 9.
2. Convertir le tableau de caractères en un tableau d'entiers et afficher ce tableau.
3. Convertir le nombre entré sous forme de chaîne de caractères en nombre entier et l'afficher.

### Exercise 3 :

1. Saisir un nombre entier positif.
2. Convertir ce nombre en une chaîne de caractères.
3. Afficher la chaîne convertie.

### Exercise 4 :

1. Saisir deux chaînes de caractères.
2. Comparer, caractère par caractère, ces deux chaînes et déterminer si elles sont identiques ou leur ordre lexicographique.

### Exercise 5 :

1. Écrire les fonctions suivantes :
  - Conversion d'une chaîne de caractères minuscules en majuscules.
  - Extraction d'une sous-chaîne à partir de la position et de sa longueur.
  - Concaténation de deux chaînes.

### Exercise 6 :

1. Écrire un programme qui vérifie si une chaîne simple (sans espace) introduite au clavier est un palindrome. Rappel : un palindrome est un mot qui reste le même qu'on le lise de gauche à droite ou de droite à gauche (par exemple, PIERRE n'est pas un palindrome, alors que OTTO est un palindrome).

## Exercice 7 : Bonus

Refaire les exercices précédents avec les fonctions de la bibliothèque «`stdlib.h`» et «`string.h`».

## Exercice 8 : Bonus

On se propose de réaliser un programme permettant de rechercher un caractère (lettre ou chiffre ASCII) se trouvant dans une chaîne de caractères.

1. La chaîne de caractères sera stockée dans un tableau de char. Le caractère à rechercher sera dans une variable de type char. On passera ces deux données à une fonction de recherche.
2. Améliorer le programme en permettant à un utilisateur d'entrer une chaîne de caractères et le caractère à rechercher. On supposera que la chaîne de caractères entrée ne dépasse pas 50 caractères. Le programme affichera le nombre d'occurrences du caractère dans la chaîne de caractères.

## ANNEXE : Compléments sur les chaînes de caractères

**Définition d'une chaîne de caractères** Une chaîne de caractères est un tableau à une dimension dont le dernier élément (caractère) est suivi de la marque de fin de chaîne `'\0'`. La taille maximale du tableau déterminée dans l'algorithme sera donc à augmenter de 1 dans le programme C. Lors de la saisie d'une chaîne de caractères, il ne faudra pas oublier de rajouter `'\0'` dans le tableau après le dernier caractère. D'autre part, cette marque de fin de chaîne sera à exploiter dans les fonctions de traitement de chaîne.

### Exemple d'utilisation

```

1  #include <stdio.h>      // a utiliser pour getchar, putchar
2  #define FINCH '\n'
3  #define TAILLEMAX 31
4  void main()
5  {
6      char chcar [TAILLEMAX] ;
7      int C;
8      char *adch;
9
10     adch = chcar;
11     C = getchar();
12     while (( C != FINCH ) && ( adch < ( chcar+TAILLEMAX-1)))
13     {
14         *adch = C;
15         adch++;
16         C = getchar();
17     }
18     *adch = '\0';
19     printf ( ' '%s' ', chcar ),
20     C=getchar() ;
21 }
22

```

- **getchar()** : récupération du caractère suivant compris dans une chaîne saisie au clavier. Attention : cette fonction est de type int.
- **putchar(nom)** : affiche le caractère nom (une constante ou une variable) à l'écran.
- **islower(nom)** : renvoie une valeur non nulle si le caractère nom est une minuscule.
- **isupper(nom)** : renvoie une valeur non nulle si le caractère nom est une majuscule.
- **toupper(nom)** : convertit le caractère nom en majuscule.
- **tolower(nom)** : convertit le caractère nom en minuscule.
- **strcpy(s,t)** : recopie la chaîne t dans la chaîne s.
- **strcmp(s,t)** : renvoie 0 si s est identique à t.