

## Chaînes de caractères Travaux Pratiques – Séance n° 7

En C, il n'existe pas de type chaîne de caractères spécifique. Les chaînes de caractères sont des tableaux de caractères, mais le dernier caractère doit être le délimiteur de fin de chaîne `'\0'`. Ainsi, `sizeof "abcdefghi"` est égal à 10 et non pas 9.

Une constante chaîne de caractères peut être définie par une suite de caractères délimitée par des guillemets, ou à l'aide d'une initialisation classique de tableau. Le caractère `'\0'` est automatiquement ajouté à la fin d'une constante littérale. Le caractère `<saut de ligne>` permet d'écrire la constante sur plusieurs lignes.

```
char str1 [10] = {'a', 'b', 'c', '\0'};
char str2 [10] = "abc";
"Une chaîne de caractères"
"" /* une chaîne vide */
" \
  ?" /* une chaîne sur plusieurs lignes */
```

Contrairement à d'autres langages, il n'y a aucun opérateur spécifique pour les chaînes de caractères en C. La bibliothèque C contient de nombreuses fonctions de manipulation de chaînes de caractères. Les prototypes des fonctions sont définis dans `string.h`. Faites un `man string.h` pour avoir une idée précise. Il est important de noter que ces fonctions ne gèrent pas l'allocation de la mémoire pour les caractères de chaînes (sauf `strdup`).

1) Écrivez et testez le programme suivant :

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main(void) {
    char str1 [10] = {'c', 'o', 't', '\0'};
    char str2 [10] = "abri";

    printf("(%s) : (%s)\n", str1, str2);
    printf("(%c) : (%c)\n", str1[1], str2[2]);
    printf("(%lu) : (%lu)\n", sizeof str1, sizeof "abc");
    printf("(%zd) : (%zd)\n", strlen(str1), strlen(str2));
    printf("(%s)\n", strcat(str2, str1));
    return EXIT_SUCCESS;
}
```

2) Récrivez les fonctions `strlen` et `strchr`.

3) Récrivez les fonctions `strcpy` et `strcat`.

4) Écrivez une fonction `compare` qui prend deux chaînes de caractères, `s1`, et `s2`, en paramètre et les compare selon l'ordre lexicographique. La fonction renverra :

— une valeur négative, si `s1` est inférieure à `s2`;

- 0, si s1 est égale à s2 ;
- une valeur positive, si s1 est supérieure à s2.

5) Comment récrire les fonctions précédentes afin qu'elles puissent faire des vérifications sur la taille des chaînes (par exemple, pour `strcpy`, que le premier paramètre a assez de place pour accueillir tous les caractères du deuxième) ?