

Fiche n° 10 de TP

Chaînes de caractères

Objectifs : manipulation de chaînes de caractères ; initialisation, parcours.

Prérequis : notion de tableau, de chaînes ; longueur, indice, caractère '\0', opérateur [].

Travail minimum : exercices 1 à 4.

Exercice 1

Ouvrez le fichier `concat_chaine.c`.

```
TP10_src/concat_chaine.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(void) {
6     char msg[] = "Bonjour";
7     strcat(msg, " tout le monde");
8
9     size_t n = strlen(msg);
10    printf("Message de longueur %zu : %s\n", n, msg);
11
12    return EXIT_SUCCESS;
13 }
```

Compilez-le. Que se passe-t-il ? Pourquoi ?

Modifiez votre programme afin qu'il affiche

```
Message de longueur 21 : Bonjour tout le monde
```

Exercice 2

Complétez le prototype et le corps de la fonction `nb_occurrences` du fichier `chaines_occurrences.c`. Cette fonction prend en paramètres une chaîne de caractères `s` ainsi qu'un caractère `c` et renvoie le nombre d'apparitions de `c` dans `s`.

Exercice 3

Complétez le corps de la fonction `remplace` du fichier `chaines_remplace.c`.

Cette fonction prend en paramètres une chaîne de caractères `s` ainsi que deux caractères `c1` et `c2`. Elle modifie la chaîne `s` en remplaçant toutes les occurrences de `c1` par `c2`.

Comparez cette fonction avec celle écrite dans l'exercice 2. Pourquoi ne faut-il pas écrire `const` dans les paramètres de la fonction `remplace` ?

Exercice 4

Inspirez vous des exercices 2 et 3 pour répondre aux questions suivantes :

- 1) Écrivez une fonction `void str_cat(char *debut, const char *ajout)` qui concatène `ajout` à la suite de `debut`.
- 2) Écrivez une fonction `void str_ncat(char *debut, const char *ajout, size_t n)` qui place au plus `n` caractères de `ajout` à la suite de `debut`.
- 3) Écrivez un programme permettant de tester vos deux fonctions.

Exercice 5

- 1) Écrivez une fonction qui supprime un caractère donné d'une chaîne de caractères passée en paramètres.
- 2) Écrivez un programme permettant de la tester.

