

Exercice 1 : Tableaux

Pour les exemples ci-dessous, indiquez successivement s'ils compilent sans erreur et le cas échéant, si les chaînes de caractères sont modifiables ou non.

- (a) `char s[] = "foo";`
- (b) `char s[42] = "Les carottes sont cuites.";`
- (c) `char s[2] = "Babel Fish";`
- (d) `char s[] = "foo\0bar\0";`
- (e) `char *s = "Prix Béton 1981";`
- (f) `char *s[] = {"foo", "bar", "baz"};`
- (g) `char s[] = {"foo" "bar" "baz"};`

Exercice 2 : Manipulation des chaînes de caractères

Que retournent chacune de ces fonctions ?

- (a) `int foo() { char s[] = "foo"; return strlen(s); }`
- (b) `int foo() { char s[] = "foo"; return sizeof(s); }`
- (c) `int foo() { char s[] = "foo"; return s[3]; }`
- (d) `int foo() { char s[] = "foo"; return s[4]; }`
- (e) `int foo() { char s[] = "foo\n\t\0bar"; return strlen(s); }`

Exercice 3

En utilisant la bibliothèque `ctype.h`, écrire une fonction qui transforme une chaîne de caractères reçue en majuscules. Le prototype de la fonction est `void majuscule(char *s)`.

Exercice 4

Soit un texte d'entrée, indiquer la position à laquelle se trouve une sous-chaîne, si elle existe. Sinon retourne -1. Le prototype de la fonction est `int position(char s[], char sub[])`. Aidez-vous également de la bibliothèque `ctype.h`.

Exercice 5

Que fait cette fonction et que retourne-t-elle ?

```
int mysterious()
{
    char v[] = "aeiouy";
    char s[] = "Hello, world!";
```

```
size_t c = 0, k = 0;
while (s[k] != '\0')
{
    int n = 0;
    while (v[n] != '\0')
        c += s[k] == v[n++];
    k++;
}
return c;
}
```

Exercice 6 : Problème Difficile

Il existe trois types de modifications qui peuvent être appliquées à une chaîne de caractère : insérer un caractère, supprimer un caractère, ou remplacer un caractère. Considérant deux chaînes de caractères, écrivez une fonction qui retourne si les deux chaînes ne sont qu'à une modification près.

- pale, ple -> true
- pales, pale -> true
- pale, bale -> true
- pale, bake -> false