

## Programmation en Langage C

### TD n°4

#### Exercice 1 :

Écrire une fonction `char * recherche_char(char* s, char c)` qui renvoie un pointeur vers la première occurrence du caractère `c` (le caractère passé en argument, pas le caractère `'c'`) dans la chaîne `s`. Si ce caractère n'apparaît pas dans la chaîne, la fonction devra renvoyer `NULL`

#### Exercice 2 :

À l'aide de `recherche_char`, écrire une fonction `int compte_char(char *s, char c)` qui compte le nombre d'occurrences de `c` dans `s`.

#### Exercice 3

Ecrire un programme qui lit 5 mots, séparés par des espaces et qui les affiche ensuite dans une ligne, mais dans l'ordre inverse. Les mots sont mémorisés dans 5 variables `M1`, ... ,`M5`.

##### Exemple

```
voici une petite phrase !  
! phrase petite une voici
```

#### Exercice 4

Ecrire un programme qui lit une ligne de texte (ne dépassant pas 200 caractères) la mémorise dans une variable `TXT` et affiche ensuite:

- a) la longueur `L` de la chaîne.
- b) le nombre de `'e'` contenus dans le texte.
- c) toute la phrase à rebours, sans changer le contenu de la variable `TXT`.
- d) toute la phrase à rebours, après avoir inversé l'ordre des caractères dans `TXT`:

```
voici une petite phrase !  
! esarhp etitep enu iciov
```

#### Exercice 5

Ecrire un programme qui lit un texte `TXT` (de moins de 200 caractères) et qui enlève toutes les apparitions du caractère `'e'` en tassant les éléments restants. Les modifications se feront dans la même variable `TXT`.

##### Exemple:

```
Cette ligne contient quelques lettres e.  
Ctt lign contint qulqus lttrs .
```

#### Exercice 6

Ecrire un programme qui lit un nombre entre 1 et 7 et qui affiche le nom du jour de la semaine correspondant:

```
"lundi"    pour 1
"mardi"    pour 2
...
"dimanche" pour 7
```

Utiliser le premier élément du tableau pour mémoriser un petit message d'erreur.

## Exercice 7

Ecrire un programme qui lit 5 mots, séparés par des espaces et qui les affiche ensuite dans une ligne, mais dans l'ordre inverse. Les mots sont mémorisés dans un tableau de chaînes de caractères.

### Exemple

```
voici une petite phrase !
! phrase petite une voici
```

## Exercice 8

Ecrire un programme qui demande l'introduction du nom et du prénom de l'utilisateur et qui affiche alors la longueur totale du nom sans compter les espaces. Employer la fonction **strlen**.

### Exemple:

```
Introduisez votre nom et votre prénom:
Mickey Mouse

Bonjour Mickey Mouse !
Votre nom est composé de 11 lettres.
```

## Exercice 9

Ecrire un programme qui lit deux chaînes de caractères CH1 et CH2, les compare lexicographiquement et affiche le résultat:

### Exemple:

```
Introduisez la première chaîne: ABC
Introduisez la deuxième chaîne: abc
"ABC" précède "abc"
```

## Exercice 10

Ecrire un programme qui lit deux chaînes de caractères CH1 et CH2 et qui copie la première moitié de CH1 et la première moitié de CH2 dans une troisième chaîne CH3. Afficher le résultat.

- a) Utiliser les fonctions spéciales de *<string>*.
- b) Utiliser uniquement les fonctions **gets** et **puts**.

## Exercice 11

Ecrire un programme qui lit un verbe régulier en "er" au clavier et qui en affiche la conjugaison au présent de l'indicatif de ce verbe. Contrôlez s'il s'agit bien d'un verbe en "er" avant de conjuguer.

Utiliser les fonctions **gets**, **puts**, **strcat** et **strlen**.

### Exemple:

```
Verbe : fêter
je fête
tu fêtes
il fête
nous fêtons
vous fêtez
ils fêtent
```

## Exercice 12

Lesquelles des chaînes suivantes sont initialisées correctement ? Corrigez les déclarations fausses et indiquez pour chaque chaîne de caractères le nombre d'octets qui sera réservé en mémoire.

- a) `char a[] = "un\ndeux\ntrois\n";`
- b) `char b[12] = "un deux trois";`
- c) `char c[] = 'abcdefg';`
- d) `char d[10] = 'x';`
- e) `char e[5] = "cinq";`
- f) `char f[] = "Cette " "phrase" "est coupée";`
- g) `char g[2] = {'a', '\0'};`
- h) `char h[4] = {'a', 'b', 'c'};`
- i) `char i[4] = "o";`

## Exercice 13

Soient les instructions:

```
char STR[200];  
puts("Entrez un nombre :");  
gets(STR);  
printf("Entrée = %s \n", STR);  
printf("integer = %d \n", atoi(STR));  
printf("long = %ld \n", atol(STR));  
printf("double = %f \n", atof(STR));
```

Quelles sont les valeurs affichées si on entre les chaînes de caractères suivantes:

- a) 123
- b) -123
- c) - 123
- d) 123.45
- e) 12E3
- f) 1234f5
- g) -1234567
- h) 123e-02
- i) -0,1234