

TP N°3

Les chaînes de caractères

Objectifs :

- ✓ Saisie et écriture de chaînes de caractères
- ✓ Manipulation de chaînes
- ✓ Allocation de mémoire statique et dynamique

Préambule

Après avoir ouvert virtualBox et vous être logué sous l'environnement CentOS5, on vous demande de créer un répertoire **tp3** dans votre répertoire de travail **algo**. Vous vous placerez ensuite dans le répertoire **tp3**.

1. Recherchez les voyelles dans un texte (TD5)

Nombre de voyelles

Ecrivez une fonction `nombreVoyelles()` qui prend en paramètre une chaîne de caractères et qui retourne le nombre de voyelles contenues dans la chaîne. Vous sauvegarderez cette fonction dans le fichier `voyelles.c` auquel vous associerez un fichier d'en-tête `voyelles.h`.

Dans un fichier que vous nommerez `tp3_exo1.c`, vous écrirez un programme principal qui demande à l'utilisateur de saisir une chaîne de caractères puis qui affiche le nombre de voyelles trouvées dans la chaîne précédente en appelant la fonction `nombreVoyelles()`.

```
> Entrez une chaîne de caractères (<50) :  
> Bonjour ca va ?  
>  
> Il y a 5 voyelles !
```

Affichez les voyelles

Ecrivez une fonction `recupVoyelles()` qui prend en paramètre une chaîne de caractères et qui retourne une chaîne de caractères contenant les voyelles trouvées. Vous sauvegarderez cette fonction également dans le fichier `voyelles.c` en pensant à compléter le fichier d'en-tête `voyelles.h`.

Dans le fichier `tp3_exo1.c`, vous modifierez le programme principal afin qu'il affiche les voyelles trouvées dans la chaîne saisie par l'utilisateur en appelant la fonction `recupVoyelles()`.

```
> Entrez une chaîne de caractères (<50) :  
> Bonjour ca va ?  
>  
> Il y a 5 voyelles !  
> Les voyelles sont : oouaa
```

2. Conjugaison d'un verbe

Le but de l'exercice est de faire conjuguer (à différents temps) un verbe du 1^{er} groupe. Dans ce but on va demander à l'utilisateur de saisir :

- Un pronom personnel (*je, tu, il, elle, nous, vous, ils, elles*),
- L'infinitif d'un verbe régulier du 1^{er} groupe (ex : chanter, danser,...)

En fonction des éléments saisis par l'utilisateur votre programme doit conjuguer correctement.

Présent

On vous demande d'écrire une fonction `char* conjugue(char* pronom, char* verbe)` qui retourne le verbe conjugué résultant de la conjugaison au présent du pronom et du verbe passés en paramètres de la fonction. Cette fonction devra vérifier que les arguments d'entrées sont bien valides

- Pronom : *je, tu, il, elle, nous, vous, ils, elles*
- Verbe : taille supérieure à 4 et se termine par *er*

Vous sauvegarderez cette fonction dans le fichier `conjugaison.c` auquel vous associerez le fichier d'en-tête `conjugaison.h`

Vous créerez enfin un fichier nommé `tp3_exo2.c` dans lequel vous viendrez tester la fonction `conjugue()`.

```
> Veuillez saisir un pronom:  
> vous  
> Veuillez saisir un verbe du 1er groupe a l'infinitif:  
> chante  
> La conjugaison au présent donne : vous chantez
```

```
> Veuillez saisir un pronom:  
> vous  
> Veuillez saisir un verbe du 1er groupe a l'infinitif:  
> dormir  
> Le verbe n'est pas un infinitif du 1er groupe !!!
```

```
> Veuillez saisir un pronom:  
> ilss  
> Veuillez saisir un verbe du 1er groupe a l'infinitif:  
> dormir  
> Le pronom personnel ne correspond pas !!!  
> Le verbe n'est pas un infinitif du 1er groupe !!!
```

Futur et imparfait

On vous demande de modifier la fonction la fonction `conjugue()` pour qu'elle puisse aussi conjuguer à l'imparfait et au futur. Cette fonction aura le prototype suivant :

```
char *conjugue ( char *pronom, char *infinitif, char *temps );
```

où `temps` vaudra « présent », « imparfait » ou « futur ».

Vous modifierez le fichier `tp3_exo2.c` afin de tester la nouvelle fonction `conjugue()`.

```
> Veuillez saisir un pronom:  
> vous  
> Veuillez saisir un verbe du 1er groupe a l'infinitif:  
> chante  
> La conjugaison au présent donne : vous chantez  
> La conjugaison au futur donne : vous chanterez  
> La conjugaison a l'imparfait donne : vous chantiez
```