Lycée TOUCHARD WASHINGTON

Langage C

TD n°7

SNir 1
2021-2022

Création de fonctions de manipulation de chaînes de caractères

Résumé du cours

Il **n'existe pas** de type chaîne en langage C, une chaîne de caractères est :

- Un tableau de caractères
- C'est le caractère **'\0',** permet d'indiquer la fin de la chaîne.

Le prototype d'une fonction indique le type de la valeur de retour, s'il y en a une, le nom de la fonction et entre parenthèses les éventuels paramètres que la fonction reçoit. Il se termine par un point-virgule.

Exemple : typeDeRetour **NomDeLaFonction** (typeDuPremierparametre nomDuParametre,..) ;

size_t **strlen(**const char *s**);**

Avec : *Paramètre de retour de type* ***size_t***
Paramètre d'entrée : char ***s***

Le mot clé **const** signifie que le paramètre est uniquement en entrée, sinon il n'apparaît pas.

La définition d'une fonction est le code correspondant à la fonction. Il n'y a pas de point-virgule à la fin de la ligne.

Exemple :

```
typeDeRetour NomDeLaFonction (typeDuPremierparametre nomDuParametre,..)
{
        // traitement réalisé par la fonction.
        return …. ;
}
```

Le nom d'un tableau représente l'adresse de la première case du tableau lorsqu'il est passé en paramètre à une fonction, il ne faut pas mettre de **&** devant si on souhaite un paramètre de sortie. De même, on ne met pas les crochets également.

Exemple :

```
char chaine[10] ;
scanf( ''%s'', chaine) ;
```

1- Les palindromes

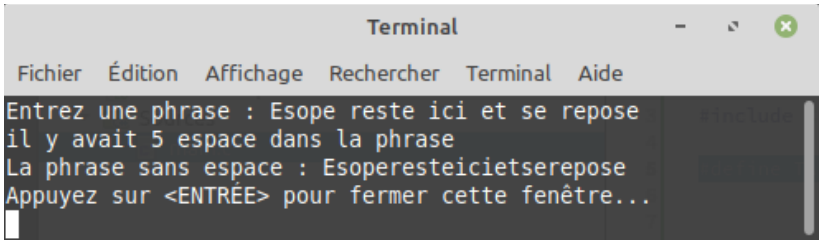
Extrait de Wikipedia

Le palindrome (adjectif et substantif masculin), du grec πάλιν / *pálin* (« en arrière ») et δρόμος / *drómos* (« chemin, voie »), aussi appelé palindrome de lettres est une figure de style désignant un mot ou une phrase dont l'ordre des lettres reste le même qu'on les lise de gauche à droite ou de droite à gauche, comme pour les mots Laval, bob ou comme dans la phrase « Ésope reste ici et se repose » ou encore « La mariée ira mal » à un accent près.

1. Créez un projet nommé **TD7_Palindrome** avec Netbeans
2. Proposez le prototype de la fonction **SupprimerEspace** qui a pour rôle de supprimer tous les espaces d'une chaine de caractères passée en paramètre. Elle retourne le nombre d'espaces supprimés, 0 s'il n'y avait pas d'espace dans la chaîne. Placez le prototype avant la fonction **main**.
3. Réalisez la définition de la fonction après la fonction **main**.
4. Complétez le programme principal pour qu'il appelle la fonction **SupprimerEspace**, indique le nombre d'espaces effectivement supprimés et affiche la phrase sans espace.

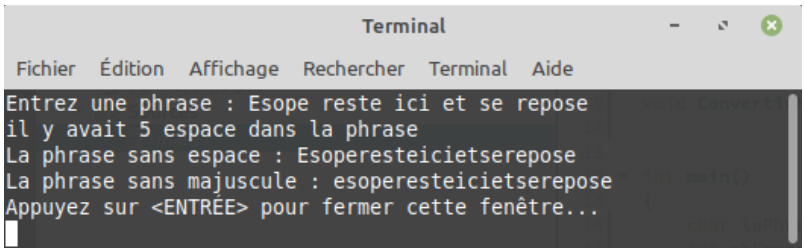
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TAILLE_CHAINE 50

// position des prototypes

int main()
{
    char laPhrase[TAILLE_CHAINE];
    int nbEspace;

    printf("Entrez une phrase : ");
    fgets(laPhrase,TAILLE_CHAINE-1,stdin);
    laPhrase[strlen(laPhrase) -1 ] = '\0';      // supprime \n à la fin de la phrase.


    // complétez à partir d'ici

    return EXIT_SUCCESS;
}

// définition des fonctions
```

Résultat attendu :



5. En utilisant le même principe pour la position du prototype et de la définition de la fonction, réalisez une nouvelle fonction **ConvertirMajusculeEnMinuscule** qui convertit toutes les majuscules d'une phrase passée en paramètre en minuscules. Cette fonction ne possède pas de paramètre de retour.

Résultat attendu :



6. Pour être complet, réalisez une fonction SupprimerAccent qui, dans les mêmes conditions que la fonction précédente, supprime les accents d'une phrase passée en paramètre.
7. Réalisez la fonction VerifierPalindrome dont le rôle est de vérifier si une phrase passée **en paramètre d'entrée** est un palindrome. Elle retourne 0, si la phrase n'est pas un palindrome et 1 si effectivement c'est un palindrome. Elle appellera au préalable toutes les fonctions nécessaires pour vérifier que la phrase est bien un palindrome.
8. Après avoir mis en commentaire votre précédent programme principal, recréez un nouveau **main** pour tester votre fonction **VerifierPalindrome** et afficher le résultat.

2- Conjugueur

Le 1er groupe comprend tous les verbes dits « réguliers » qui finissent en -er (sauf le verbe aller). Ils représentent près de 90 % des verbes. Au présent les terminaisons sont les suivantes pour un verbe xxxER on remplace ER par la terminaison adaptée :

Je	xxxE	Nous	xxxONS
Tu	xxxES	Vous	xxxEZ
Il/Elle	xxxE	Ils/Elles	xxxENT

On se propose de faire une application permettant de donner la conjugaison des verbes du premier groupe au présent. Pour cela, il faut répondre à plusieurs problématiques :

- Comment récupérer la dernière lettre d'une chaîne de caractères ?
- Comment récupérer l'avant-dernière lettre d'une chaîne de caractères ?
- Comment afficher tous les caractères d'une chaîne sauf les 2 derniers ?
- Comment savoir si le verbe saisi est le verbe aller ?

Pour le traitement du problème, vous réaliserez des fonctions. Une fonction réalise un seul traitement par exemple, rechercher si un verbe est du premier groupe.

1. Proposez un programme qui demande un verbe à l'infinitif. Si ce dernier se termine par "er" et n'est pas le verbe aller, affichez la conjugaison de ce dernier au présent de l'indicatif.

On souhaite améliorer l'application afin que si le mot commence par une voyelle, on ait "j'" et non "je" pour la première personne du singulier.

2. Proposez une solution.

Les verbes en -ger comme manger, ranger... prennent un e intercalaire à la 1ère personne du pluriel (ex: pour le verbe manger, on aura mangEons et non mangons).

3. Proposez une solution pour répondre à cette problématique.

3- Crypteur

On souhaite pouvoir crypter/décrypter des messages en utilisant un algorithme de substitution des lettres.

Voici une table de substitution proposée :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
@	8	[	0	3	{	6	#	1	}	:	7	W	Z	*	?	O	%	$	-	V	^	M	+	/	N

Le fonctionnement de l'application sera le suivant :

Via un menu, l'utilisateur pourra choisir s'il souhaite crypter ou décrypter un message. Dans les 2 cas, il lui sera demandé le message à crypter ou à décrypter. Suivant le choix qu'il aura fait précédemment, le cryptage ou le décryptage du message qu'il vient de saisir s'affichera.

Organisation du programme :

Le cryptage et le décryptage seront réalisés dans 2 fonctions indépendantes
Le programme principal assure la gestion du menu, la saisie du message, l'appel d'une deux fonctions en fonction du choix de l'utilisateur, puis l'affichage du message crypté ou décrypté suivant le cas.

Il est conseillé de déclarer 2 tableaux initialisés, l'un avec les lettres , l'autre avec les caractères de substitution correspondants. Au niveau du traitement de la chaîne saisie, il faudra commencer par mettre toutes les lettres en majuscule. Les caractères accentués ne seront pas traités.

Pour le chiffrement, il suffit de parcourir la chaîne de caractères et de tester si le caractère courant est une lettre, de calculer l'indice correspondant et de remplacer cette lettre par le code de substitution. Si le caractère courant n'est pas une lettre, on ne le remplace pas.

Exemple avec Hello world :

	H	**E**	**L**	**L**	**O**		**W**	**O**	**R**	**L**	**D**
indice de la lettre	**7**	**4**	**11**	**11**	**14**		**22**	**14**	**17**	**11**	**3**
caractère de substitution correspondant à l'indice	**#**	**3**	**7**	**7**	*****		**M**	*****	**%**	**7**	**0**

Pour déchiffrer, le principe est le même, mais l'indice du caractère courant ne peut pas être calculé. Il faut donc rechercher le caractère dans la table de substitution et grâce à l'indice de ce dernier, mettre la lettre correspondante.