

Consignes Projet ScIN1B

Vous pouvez réaliser ce travail en groupe (max. 2 étudiants) sous réserve que vous fassiez partie du même groupe de TD.

Le rapport expliquant votre code (au format *.pdf) et le code (une archive contenant les fichiers pour chaque exercice) sont à déposer sur Plubel-prod le **11/12/2021** à 23:59, au plus tard. Les noms de l'archive et du fichier PDF soumis doivent comporter vos noms de famille en majuscules, ainsi que le nom de la formation et du module (e.g. "L1_ScIN1B").

Vous devez présenter votre travail lors des séances TD programmées dans la semaine du 13/12/2021.

Les travaux soumis mais non présentés en TD se verront affecter la note de 0/20.

Exercice 1

Consignes:

Ecrivez un programme Perl qui, à partir d'un texte saisi, reconnaît un mot se terminant par la lettre "a" et l'enregistre dans **\$1**. Affichez ensuite le texte précédant le mot reconnu, le mot reconnu entre "<" et ">" et le reste du texte saisi. Affichez ensuite soit le contenu de **\$1**, soit un message si rien n'a été trouvé.

Exemple d'exécution:

```
Veillez saisir votre texte - terminez votre saisie en appuyant sur ENTREE - quittez le programme avec Ctrl + Z
>Tentative d'essai pour détecter un mot qui se termine en a et rien d'autre
Mot reconnu: |Tentative d'essai pour détecter un mot qui se termine en <a> et rien d'autre|
$1 contient 'a'
Veillez saisir votre texte - terminez votre saisie en appuyant sur ENTREE - quittez le programme avec Ctrl + Z
>Recherche de mots se terminant par la lettre a et rien d'autre
Mot reconnu: |Recherche de mots se terminant par <la> lettre a et rien d'autre|
$1 contient 'la'
Veillez saisir votre texte - terminez votre saisie en appuyant sur ENTREE - quittez le programme avec Ctrl + Z
>
Aucun mot reconnu: ||
```

Indications complémentaires:

Vous pouvez utiliser l'entre standard **<STDIN>** pour récupérer ce que l'utilisateur saisit au clavier. **STDIN** est un descripteur fourni par le système d'exploitation, permettant de lire l'entrée du clavier. Plus de détails: <https://www.commentcamarche.net/contents/777-perl-entree-et-sortie-standards>

Vous pouvez utiliser l'opérateur **chomp** pour supprimer le caractère de nouvelle ligne "\n" dans une chaîne de caractères (CM03 diapo 67):

- `$text = "une ligne de texte\n"; # ou <STDIN>`
- `chomp($text); #supprime \n du contenu de la variable $text`

Exercice 2

Fichier source: Exo2.txt (fourni sur Plubel-prod)

Consignes:

Ecrivez un programme Perl qui copie un fichier puis en modifie le contenu. Lors de la modification, chaque occurrence de la chaîne "**Fred**" (insensible à la casse) devra être remplacée par la chaîne

"Larry". Le nom du fichier à copier doit être fourni en tant qu'argument. Le nom du fichier copié et modifié doit être le même que celui du fichier d'origine et il faut lui ajouter l'extension ".out".

Si le fichier d'origine ne contient pas d'extension, on ajoute ".out". Si le fichier d'origine contient une extension, alors celle-ci est remplacée par ".out".

Affichez le nom du fichier généré et le nombre de remplacements effectués.

Exemple d'exécution:

```
$ ./Exo2.pl Exo2.txt
Fichier généré: 'Exo2.out'
Remplacements effectués: '11'
```

Exercice 3

Consignes:

Ecrivez un programme Perl qui va demander à l'utilisateur de deviner un nombre entre 1 et 100. Tant que l'utilisateur n'a pas deviné le nombre correct, le programme continue de lui demander de saisir un nombre. Pour chaque nombre saisi, le programme affiche "Trop petit" ou "Trop grand" selon le nombre à deviner. Si l'utilisateur saisit "quitter" (insensible à la casse), ou une ligne vide, le programme doit quitter et afficher un message à l'utilisateur. Si l'utilisateur devine, le programme affiche un message à l'utilisateur et quitte.

Indications complémentaires:

Nous avons vu en cours l'opérateur **next** (CM03 diapo 55). Dans cet exercice, vous pouvez utiliser l'opérateur **last** qui termine immédiatement l'exécution d'une boucle (équivalent du "**break**" dans le langage C par exemple). L'opérateur **last** peut être utilisé avec tous les types de boucles Perl que nous avons vues: **for**, **foreach**, **while** et **until**. Plus de détails: <https://formation-perl.fr/guide-perl-03.html>

Code exemple	Commentaires
<pre>while (condition) { if (/FIN/) { # On veut quitter la boucle last; } elsif (/toto/) { print; } } ## last permet d'arriver ici ##</pre>	<p>L'opérateur last s'applique à toute la boucle de contrôle, pas uniquement au test if.</p>

Exemple d'utilisation:

```
$ ./Exo3.pl
Veuillez saisir un nombre entre 1 et 100:: 50
Trop grand. Veuillez réessayer!
Veuillez saisir un nombre entre 1 et 100:: 40
Trop grand. Veuillez réessayer!
Veuillez saisir un nombre entre 1 et 100:: 20
Trop grand. Veuillez réessayer!
Veuillez saisir un nombre entre 1 et 100:: 15
Trop petit. Veuillez réessayer
Veuillez saisir un nombre entre 1 et 100:: 17
Trop petit. Veuillez réessayer
Veuillez saisir un nombre entre 1 et 100:: 18
Et voilà!
```

Exercice 4

Consignes:

Ecrivez un programme Perl qui, pour un nombre saisi par l'utilisateur, affiche tous ses diviseurs (sauf 1 et le nombre lui-même). Par exemple, pour le nombre 99, le programme doit afficher les diviseurs 3, 9, 11, 33. Si le nombre n'a pas de diviseurs, le programme doit afficher qu'il est premier. Si l'utilisateur ne saisit pas un nombre, le programme l'indique à l'utilisateur, sans essayer de déterminer des diviseurs.

Indications complémentaires:

Vous devez utiliser la fonction suivante pour retourner une liste de diviseurs:

```
sub diviseurs {  
    my $nb = shift;  
    my @diviseurs = ();  
    foreach my $diviseur (2 .. ($nb/2)) {  
        push @diviseurs, $diviseur unless $nb % $diviseur;  
    }  
    return @diviseurs;  
}
```

Exemples d'exécution:

```
$ ./Exo4.pl 17  
Vérification du nombre <17>  
17 est un nombre premier
```

```
$ ./Exo4.pl 135  
Vérification du nombre <135>  
135 est divisible par 3 5 9 15 27 45
```

```
$ ./Exo4.pl 1368745423  
Vérification du nombre <1368745423>  
1368745423 est divisible par 5281 259183
```

```
$ ./Exo4.pl 11abc  
Vérification du nombre <11abc>  
Vous n'avez pas saisi un nombre
```