Travaux dirigés de Perl n°1/6

Scalaires, tableaux, fonctions

—Université Gustave Eiffel—

Le but de ce TD est l'apprentissage du langage *Perl* par quelques exemples simples de sa syntaxe : expressions, boucles, listes ...
Toutes les fonctions écrites devront être testées.

▸ Exercice 1.

De quelle version de Perl disposez-vous ?

Quelles furent les options de compilation de cet interpréteur ?

▸ Exercice 2.

Copier le fichier `/home/ens/lhullier/ens/mystere.pl` *ou* `https://formation-perl.fr/t/mystere.pl`
Que fait ce programme ? Modifiez le pour qu'il fonctionne correctement

```
#!/usr/bin/perl
my $nntpmntp = 10;
while($nntpnmtp>5) {
   print "$nntpmntp\n";
   $nntpmntp --;
}
```

```
my $x = 'oui';
my $y = 'non';
if( $x == $y )
   print "c'est dingue!\n";
else
   print "tout va bien\n";
print '2'.'9'+'5'."\n";
```

▸ Exercice 3.

Écrivez une fonction `SommeTest` *qui prend trois paramètres* `x`, `y` *et* `n` *et qui renvoie vrai ou faux selon si la somme de* `x` *avec la concaténation de* `x` *et* `y` *vaut* `n` *ou non (vrai ou faux au sens booléen de Perl). Par exemple* `SommeTest(2,10,212)` *renvoie vrai.*

Comment tester cette fonction ? Comment utiliser les paramètres de la ligne de commande ?

▸ Exercice 4.

Écrivez les fonctions suivantes qui prennent un paramètre numérique `n` *:*

. `TableMult1` *qui affiche la table de multiplication carrée de 1 à* `n` *en utilisant* `for(;;)`
Les colonnes seront alignées sur 5 caractères : utilisez `printf('%5d',...)`

Voici la table pour `n` *= 4 :*

1	*2*	*3*	*4*
2	*4*	*6*	*8*
3	*6*	*9*	*12*
4	*8*	*12*	*16*

. `TableMult2` *qui affiche la table de multiplication de 1 à* `n` *en utilisant* `foreach`

. `TableMult3` *qui renvoie une chaîne de caractères contenant la table de multiplication de 1 à* `n`.
Utilisez `sprintf` *et consultez sa documentation avec* `perldoc -f sprintf`

Votre programme prendra comme valeur le premier paramètre passé en ligne de commande.
Comment avoir une valeur par défaut ? (par exemple 10)

▸ **Exercice 5.**

- *Écrivez une fonction* `Fact` *qui calcule récursivement la factorielle de son paramètre. Appliquez cette fonction à tous les entiers de 0 à 10.*

$$\begin{cases} F_0 &= 1 \\ F_n &= n \times F_{n-1} \end{cases}$$

- *Écrivez une fonction* `Fibo` *qui calcule* **itérativement** *et renvoie tous les nombres de Fibonacci d'indice inférieur ou égal à n. Utilisez un tableau pour stocker les valeurs successives (la case d'indice i contiendra le* i^{eme} *nombre de Fibonacci).*

$$\begin{cases} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \end{cases}$$

Dans le programme principal, comment afficher tout le tableau ? La dernière valeur ?

▸ **Exercice 6.** *Écrivez un programme qui successivement :*

- *crée un tableau contenant 4, -5 et 7,*
- *ajoute -2 et 3 à la fin du tableau,*
- *affiche le tableau (les éléments seront séparés par des chaînes virgule-espace),*
- *ajoute 0 et -1 au début du tableau,*
- *remplace la valeur d'indice 3 par 9,*
- *multiplie par 2 chaque terme du tableau (fonction* `map`*),*
- *ne garde dans le tableau que les termes strictement positifs (fonction* `grep`*),*
- *trie le tableau en ordre décroissant (fonction* `sort`*).*

▸ **Exercice 7.**

Écrivez une fonction `Intervalle(`n`,`x`)` *qui renvoie la liste des nombres entiers contenus dans l'intervalle de 1 à n, duquel on enlève x (n et x entiers quelconques). Par exemple* `Intervalle(10,4)` *renvoie la liste 1,2,3,5,6,7,8,9,10.*

Écrivez une fonction `NonMult(`n`,`x`)` *qui renvoie une liste contenant l'intervalle de 1 à n privé des nombres multiples de x.*

▸ **Exercice 8.**

Écrivez une fonction qui construit la liste de tous les nombres premiers inférieurs à son paramètre ; la méthode utilisée sera un crible d'Eratosthène simplifié. **Demandez à l'enseignant**.

▸ **Exercice 9.**

Écrivez une fonction `Modif($texte,$ancien,$nouveau)` *qui remplace toutes les occurrences de la chaîne* `$ancien` *dans la chaîne* `$texte` *par la chaîne* `$nouveau`.

Pour cela, vous utiliserez les fonctions `index` *et* `substr`.

Par exemple `Modif( 'bonjour vous, bonjour', 'bonjour', 'allo');`
renvoie `'allo vous, allo'`

Vérifiez que `Modif( 'bonjour vous, bonjour', 'bonjour', 'bonjour bonjour');`
renvoie bien `'bonjour bonjour vous, bonjour bonjour'`