

Module Programmation Internet.

TD 3

Patrick Poulingeas

Exercice 1.

Ecrire un script Perl qui affiche la somme des arguments passés en ligne de commande. Si aucun argument n'est fourni au script, on affiche 0. Le script ne contrôlera pas le fait que les arguments soient des nombres ou pas.

Exercice 2.

Ecrire un script Perl qui affiche le nombre de lettres et le nombre de chiffres du premier argument qui lui est donné comme paramètre par la ligne de commande. S'il n'y a aucun argument, on affiche un message d'erreur.

Exercice 3.

Ecrire un script Perl qui attend la saisie d'une chaîne au clavier puis indique si celle-ci est un palindrome ou non.

Exercice 4.

Ecrire un script Perl qui attend la saisie d'un nombre n au clavier puis affiche les puissances de 2^0 à 2^n . On ne contrôlera pas que les caractères saisis sont tous des chiffres, ni que le nombre est un entier positif.

Exercice 5.

Ecrire en Perl une fonction qui calcule le plus grand des nombres qui lui sont passés en paramètres. On ne vérifiera pas que les paramètres sont des nombres. S'il n'y a aucun argument, la fonction retourne « undef ».

Exercice 6.

Ecrire une fonction Perl qui retourne la liste des entiers de 2^0 à 2^n , où n est le premier argument de la fonction. S'il n'y a aucun argument, on retourne la liste (2^0).

Exercice 7.

Ecrire en Perl une fonction « remplacer » qui remplace une sous-chaîne par une autre dans une chaîne (On ne remplace que la 1^{ère} occurrence de la sous-chaîne).

Exemple de résultat souhaité :

```
$s = "Le chat mange";
```

```
remplacer ($s,"chat","chien");      # $s vaut "Le chien mange"
```

Si la sous-chaîne à remplacer n'existe pas dans la chaîne, la chaîne est inchangée.

Remarque : On n'utilisera pas d'expression régulière pour écrire la fonction.

Exercice 8.

On cherche à modéliser des matrices avec des tables de hachage.

Par convention, $M = \begin{pmatrix} 3,14 & 0 \\ 0 & 1,1 \end{pmatrix}$ est représentée par :

```
my %M = ( "1:1" => 3.14,  
          "1:2" => 0,  
          "2:1" => 0,  
          "2:2" => 1.1 );
```

- a. Ecrire une fonction « creer_identite » qui, à partir d'un paramètre $n \geq 1$, retourne la matrice identité I_n dans $\mathfrak{R}^{n \times n}$. On ne contrôlera pas la validité du paramètre fourni à la fonction.
- b. Ecrire une fonction qui affiche le contenu d'une matrice. Pour la matrice M donnée précédemment en exemple, on aura l'affichage suivant :
Matrice(1,1) = 3.14
Matrice(1,2) = 0
Matrice(2,1) = 0
Matrice(2,2) = 1.1

On respectera l'ordre sur les indices apparaissant dans l'exemple ci-dessus.

On ne vérifiera pas la cohérence de la table de hachage passée en paramètre à la fonction (Par le terme « cohérence », nous entendons le fait que la table de hachage suive bien la convention de représentation choisie).