

**Rappels****(I) Variables :**

Considérons les variables \$Licences = 3 ; \$Masters = 2 ;

Afficher, à l'aide d'une seule instruction PHP, la phrase suivante :

J'ai 3 Licences et 2 Masters, j'ai donc 5 **formations**.

Les nombres 3, 2 et 5 dépendent des valeurs des variables données précédemment. Le faire une fois en utilisant uniquement les apostrophes et une fois en utilisant uniquement les guillemets.

**(II) Conditions, Boucles et Tableaux :**











Considérons par exemple les variables suivantes :

\$Ville = 'Tanger' ;

\$TempVille = 23 ;

\$TabCouleurs = array('Rouge foncé', 'Rouge', 'Ecarlate', 'Orange', 'Jaune', 'Vert riche', 'Vert', 'Bleu ciel', 'Bleu', 'Violet', 'Violet rose', 'Magenta') ;

Afficher, à l'aide d'une boucle et des instructions **if...else...elseif** et **switch**, la table HTML suivante :

Couleur	Température	Ville
	Entre 50 et 33	
	Entre 28 et 32	
	Entre 27 et 22	Tanger
	Entre 21 et 17	
	Entre 16 et 11	
	Entre 10 et 5	
	Entre 4 et 0	
	Entre -1 et -6	
	Entre -6 et -11	
	Entre -12 et -17	
	Entre -18 et -22	
	Entre -23 et -50	

L'affichage dans la table dépend des valeurs des variables données précédemment.

**(III) Fonctions :**

- (a) Ecrire une fonction **NomComple**t qui prend en paramètre deux chaînes de caractères à partir de deux zones de texte **Nom** et **Prenom** puis construit et retourne une chaîne de caractères représentant le nom complet dans une troisième zone de texte.

Exemple :

Nom :       Prenom :   
 Nom Complet :

- (b) Ecrire une fonction **Remplacer** qui prend en paramètre une chaîne de caractères **TXT** et deux lettres **X** et **Y** puis retourne une chaîne de caractères **RES** où toutes les apparitions du caractère **X** sont remplacées par **Y**.

Exemple :

TXT :       X :       Y :   
 RES :

- (c) Ecrire une fonction **Rpetition** qui prend en paramètre une chaîne de caractères **TXT** et un entier **i** puis retourne le nombre de répétitions continues dans **TXT** à partir de l'indice **i**.

**Exemple 1 :**

**TXT :** bbbbbwwwwwwwwwwwwwbbbw

**i :** 5

**nbr :** 13

**Exemple 2 :**

**TXT :** bbbbbwwwwwwwwwwwwwbbbw

**i :** 21

**nbr :** 1

- (d) Ecrire une fonction **Entier** qui prend en paramètre une chaîne de caractères **TXT** et un indice **i** puis retourne l'entier **RES** correspondant à tous les chiffres continus à partir de la case d'indice **i** ainsi que **nbr** le nombre de chiffres lus.

**Exemple 1 :**

**TXT :** 19abc386cd47

**i :** 5

**RES :** 386

**nbr :** 3

**Exemple 2 :**

**TXT :** 19abc386cd47

**i :** 2

**RES :** 0

**nbr :** 0

- (e) Afin d'économiser l'espace de stockage, on utilise souvent des algorithmes de compression. Parmi les algorithmes les plus utilisés on trouve l'algorithme de codage **RLE** vu la facilité de le programmer.

Le principe est simple, au lieu d'écrire :

**wwwwwwwwwwwwbaaaabbbcccccbww.**

Il vaut mieux écrire : **12w1b4a3b6c1b2w.**

C'est-à-dire remplacer chaque répétition d'un caractère par le nombre de répétitions suivi du caractère répété.

Ecrire une fonction **CodageRLE** qui reçoit en paramètres une chaîne de caractères **TXT** et retourne son codage **RLE**. On suppose que la chaîne **TXT** ne contient pas de chiffres.

**Exemple :**

**TXT :** bbbbbbbbbbwbbwww

**RLE :** 10b1w2b5w

- (f) Ecrire une fonction **DecodageRLE** qui reçoit en paramètres une chaîne de caractères codée en RLE et retourne le texte décodé **TXT**.

**Exemple :**

**RLE :** 12w3b1w5b

**TXT :** wwwwwwwwwwwwbbbwbbbbb

## (IV) Les fichiers :

- (a) Ecrire une fonction **creerFichier** qui reçoit en paramètre un nom de fichier **ch** reçu d'une zone de texte. Ensuite cette fonction permet d'écrire dans ce fichier le texte suivant d'une autre zone de texte :

Non du fichier :

Contenu :

ENS
DUT
Licence
Master

- (b) Ecrire une fonction **afficherFichier** qui permet de lire et d'afficher le texte déjà écrit dans l'exercice précédent.

- (c) Ecrire une fonction **monFichier1(ch)** qui reçoit en paramètre un nom de fichier. Si le fichier n'existe pas, la fonction permet de créer et d'enregistrer de nouvelles lignes. Sinon, la fonction sert à afficher le contenu du fichier.

- (d) Ecrire une fonction **monFichier2(ch1, ch2)** qui lit le contenu du fichier **ch1** et le recopie dans un autre fichier **ch2**.

- (e) Ecrire une fonction **Calculer1(ch)** qui calcule et affiche le nombre de voyelles, et le nombre de consonnes d'un fichier texte.

- (f) Ecrire une fonction **Calculer2(ch)** analysant un fichier texte et indiquant le nombre d'occurrence de caractères et de chiffres.

- (g) Ecrire une fonction **Calculer3(ch)** qui calcule le nombre de lettres, de chiffres et de lignes d'un fichier texte

**Remarques :** Le résultat doit être classé en ordre croissant et enregistré dans un fichier **Resultat.txt**.

- (h) Ecrire une fonction **chercherMot(ch, mot)** analysant un fichier texte **ch** et indiquant le nombre d'occurrences d'un mot **mot**, donné en paramètre de la fonction dans **ch**.

- (i) Ecrire une fonction **inverserFichier()** permettant de recopier le contenu d'un fichier texte à l'envers et le mets dans un deuxième fichier  
modifier la fonction pour qu'il recopie à l'envers mot par mot.

- (j) Ecrire une fonction **chargerFichier1()** qui permet de créer un fichier texte **info.txt** comportant des informations d'une entreprise sous la forme suivante :  
<Code client> <Nom client> <Ville client>

info.txt		
1	Mohamed	Tanger
2	Fatima	Tetouan
3	Brahim	Casa
4	Jamal	Tanger
5	Ikram	Casa
6	Karima	Rabat
...	...	...

**Remarques :**

— Ces informations seront lues à partir des zones de textes.

- La fonction vérifie la taille du fichier. Si le fichier dépasse 100 Ko, un message d'erreur s'affiche.

(k) Ecrire une fonction **chercherCode**(code) permettant, à partir du fichier créé par la question précédente, de retrouver les informations correspondant à un code de client reçu comme paramètre et à partir d'une zone de texte.

(l) Ecrire une fonction **chargerFichier2**() qui charge et retourne les informations du fichier de la question (e) dans une table associative **info** de la forme suivante : **array(<Code client> => array ( <Nom client>, <Ville client> ) )**.

(m) Ecrire une fonction **compterVille**(V, info) qui reçoit une chaîne de caractères **V** à partir d'une zone de texte et la table **info** de la question précédente, puis qui compte et affiche combien d'articles sont destinés à être distribuer dans la ville **V**.

(n) Ecrire une fonction **distribuerVille**(Info) qui reçoit comme paramètre la table **Info** puis enregistre les informations des produits de chaque ville dans un fichier à part.

**Exemple** : Pour la table **info** :

info	
array( 1=>	array('Mohamed', 'Tanger'),
2 =>	array('Fatima', 'Tetouan'),
3 =>	array('Brahim', 'Casa'),
4 =>	array('Jamal', 'Tanger'),
5 =>	array('Ikram', 'Casa'),
6 =>	array('Karima', 'Rabat'),
...	...
)	

La fonction doit créer les fichiers suivants :

—	Tanger.txt			Tetouan.txt		
	1	Mohamed	Tanger	2	Fatima	Tetouan
	4	Jamal	Tanger			
—	Casa.txt			Rabat.txt		...
	3	Brahim	Casa	6	Karima	Rabat
	5	Jamal	Casa			...

(V) **PHP et MySQL :**

Soient les tables suivantes d'une base de donnée qu'on appellera **GestionNotes** :

- **Etudiants** (CNE, nom, #codeClasse) :
  - CNE : code national de l'étudiant, de type TEXT, clé primaire.
  - nom : nom de l'étudiant, de type TEXT.
  - codeClasse : donnée de type TEXT, clé étrangère.
- **Classes** (codeClasse, filiere, num) :
  - codeClasse : code définissant de manière unique une classe, de type TEXT, clé primaire.
  - filiere : donnée de type TEXT désignant la filière d'une classe.
  - num : donnée de type INTEGER désignant le numéro d'une classe.
- **Matieres** (codeMat, designation) :
  - codeMat : code définissant de manière unique une matière, de type TEXT, clé primaire.
  - designation : donnée de type TEXT désignant la désignation d'une matière.
- **Notes** (#codeMat, #CNE, note) :
  - codeMat : clé étrangère.
  - CNE : clé étrangère.
  - note : donnée de type REAL désignant la note d'un étudiant pour une matière donnée.

- (a) Créer la base de donnée **GestionNotes** avec ses tables.
- (b) En utilisant des scripts PHP d'un back office (côté administration) :
  - Alimenter les tables par des données convenables.
  - Visualiser, modifier et supprimer certains enregistrements de chaque table.
- (c) En utilisant des scripts PHP d'un front office (côté client), créer cette page bulletin :

**ROYAUME DU MAROC**  
 ....  
 ....

**Français / العربية**

Filière :   
 Code National :   
 Nom et Prénom : ....

Matière	Note	Résultat (V/NV/R)
PHP	.....	.....
.....	.....	.....
<b>Moyenne Générale</b>	.....	.....