

TP 03: LES STRUCTURES REPETITIVES : WHILE

Objectifs

- Savoir répéter des instructions

Exercices

Exercice 1 : while

En utilisant la syntaxe « while », afficher les 10000 premiers nombres entiers séparés par un tiret.

Exercice 1 bis : while

Ecrire une procédure (fonction) nommée « comptage » qui reçoit un paramètre n en entrée et qui affiche les n premiers nombres entiers séparés par un tiret.

Ecrire un programme qui demande de saisir un nombre et qui appelle la procédure en utilisant ce nombre.

Exercice 2 : Bonjour

Ecrire une procédure nommée « bonjour » qui reçoit un paramètre n en entrée et qui affiche n fois le mot « Bonjour » séparé par un saut de ligne !

Ecrire un programme qui utilise votre procédure.

Exercice 3 : Divisibilité

Ecrire une fonction « divisiblePar3 » qui dit si un nombre est divisible par 3. Cette fonction reçoit le nombre en entrée et retourne « True » s'il est divisible par 3 et « False » dans le cas contraire.

En utilisant la fonction, afficher les nombres divisibles par 3 compris entre 154 et 268 en les séparant par un tiret.

Exercice 4 : Bonne nuit

- Écrivez un programme qui demande à l'utilisateur un entier n (le nombre de moutons qu'il doit compter pour s'endormir) et pour chaque entier i entre 1 et n, affiche le message « Et i moutons ! ».
- A la fin, il doit afficher le message « Bonne nuit... ».
- Faites en sorte que la première phrase soit « Et 1 mouton ! » (sans le s à la fin de mouton).
- Affichez un message d'erreur si on entre un entier négatif et recommencez la saisie

Exercice 5 : Moyenne de notes

- En utilisant la syntaxe « while », faire saisir des notes à l'utilisateur...
La fin de la saisie se fera en saisissant une note négative.
- Améliorez votre programme pour signaler à l'utilisateur une erreur de saisie si la note est supérieure à 20.

Exercice 5 bis : Moyenne, Min et Max

Complétez votre programme précédent pour permettre la saisie de notes et afficher la moyenne, la note la plus petite et le note la plus grande.

Exercice 6 : Comptage

Ecrivez un programme qui demande à l'utilisateur d'entrer un chiffre. Cette action devra être répétée tant que l'utilisateur répondra « oui » à la question voulez-vous recommencer ?

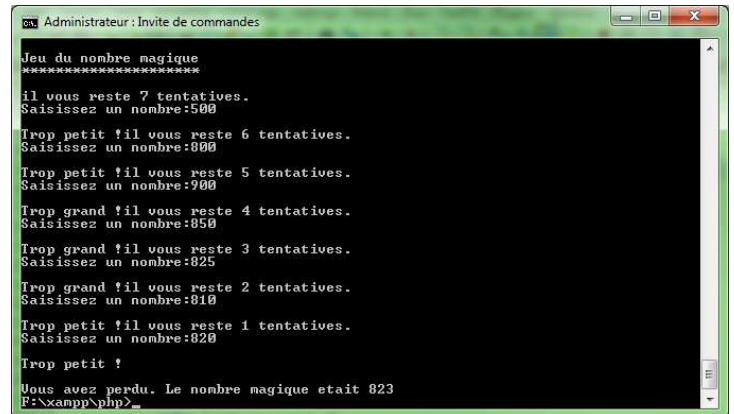
Le programme devra indiquer :

- le nombre de valeurs saisies
- le nombre valeurs paires
- le nombre valeurs impaires
- Le nombre d'erreurs de saisie

Exercice 7: Nombre magique

L'ordinateur va générer un nombre aléatoire compris entre 0 et 1000. Vous avez 7 tentatives pour le découvrir. A chaque tentative, l'ordinateur vous indique « Trop petit » ou « Trop grand »...

Pour générer un nombre aléatoire entre 0 et 1000, vous utiliserez la fonction `rand()`. Consultez la documentation de cette fonction.



```
Administrateur : Invite de commandes
Jeu du nombre magique
*****
il vous reste 7 tentatives.
Saisissez un nombre:500
Trop petit !il vous reste 6 tentatives.
Saisissez un nombre:800
Trop petit !il vous reste 5 tentatives.
Saisissez un nombre:900
Trop grand !il vous reste 4 tentatives.
Saisissez un nombre:850
Trop grand !il vous reste 3 tentatives.
Saisissez un nombre:825
Trop grand !il vous reste 2 tentatives.
Saisissez un nombre:810
Trop petit !il vous reste 1 tentatives.
Saisissez un nombre:820
Trop petit !
Vous avez perdu. Le nombre magique etait 823
F:\xampp\php>
```

Exercice 8 : Voyelles et consonnes

Saisir une phrase dans un formulaire et écrire un programme PHP qui compte le nombre de consonnes et le nombre de voyelles de cette phrase.

Pour extraire un caractère d'une chaîne, on utilisera la fonction « `substr` » (Cf ; documentation php).

Si on souhaite connaître la longueur d'une chaîne, on utilisera la fonction « `strlen` » (Cf ; documentation php).

Pour cela, vous écrirez une fonction `voyelles` qui reçoit une chaîne de caractères et retourne le nombre de voyelles. Vous écrirez aussi une fonction `consonnes` qui reçoit une chaîne de caractères et retourne le nombre de voyelles.

Les programmes qui suivent sont optionnels.

Exercice 9: Enigme

Nous allons essayer de faire résoudre cette énigme par l'ordinateur en utilisant sa capacité à calculer rapidement !

La méthode consiste à lui faire essayer toutes les possibilités et pour chacune d'elle, calculer le résultat.

1. Découverte de ce type d'énigme.

Dans un premier temps, on va tenter de résoudre une énigme très simple :

	-	2
	+	
	=	2

Comme précédemment, chaque case doit être complétée par un nombre entier entre 1 et 9.

- Chercher toutes les solutions de cette énigme à la main. Vous devriez en trouver trois, si l'on oublie que chaque chiffre ne peut être utilisé qu'une seule fois (sinon, il n'y a que deux solutions).
- Ecrire un programme contenant une double boucle et testant toutes les possibilités, sans prendre en compte la condition de non répétition des chiffres utilisés.
- Modifier le programme précédent pour que les solutions ne contiennent que des chiffres différents.

2. Tour de chauffe

On considère l'énigme suivante, plus simple que celle que l'on souhaite résoudre, mais plus compliquée que la précédente.

- Ecrire un programme permettant de résoudre cette énigme : Ce programme pourrait contenir une quadruple boucle et testera toutes les possibilités, sans prendre en compte la non répétition des chiffres utilisés.
- Modifier le programme précédent pour que les solutions ne contiennent que des chiffres différents et non nuls. Comptez le nombre de solutions trouvées.

3. L'énigme de la voix du Nord.

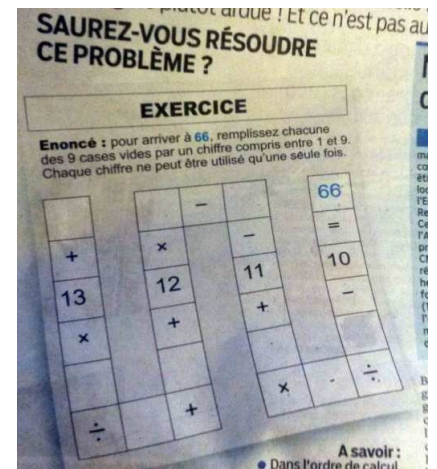
- Dans un premier temps, ne tenez pas compte de la contrainte de n'avoir que des chiffres différents. Vous devriez alors trouver beaucoup de bons résultats (Cf. exemple ci-dessous)

```
Bravo, vous avez gagné !
Les chiffres à trouver sont 1 - 1 - 3 - 4 - 7 - 7 - 2 - 3 - 9
Bravo, vous avez gagné !
Les chiffres à trouver sont 1 - 1 - 3 - 4 - 7 - 7 - 3 - 2 - 9
Bravo, vous avez gagné !
Les chiffres à trouver sont 1 - 1 - 3 - 4 - 7 - 7 - 4 - 1 - 6
Bravo, vous avez gagné !
Les chiffres à trouver sont 1 - 1 - 3 - 4 - 7 - 7 - 6 - 1 - 9
Bravo, vous avez gagné !
Les chiffres à trouver sont 1 - 1 - 3 - 4 - 7 - 8 - 1 - 5 - 3
Bravo, vous avez gagné !
Les chiffres à trouver sont 1 - 1 - 3 - 4 - 7 - 8 - 2 - 5 - 6
```

Que constatez-vous au niveau du temps de traitement ?

- Ajoutez les tests afin de s'assurer de ne pas faire le calcul lorsque 2 nombres sont identiques. Comptez le nombre de solutions trouvées !

```
Les chiffres à trouver sont 9 - 4 - 1 - 5 - 2 - 7 - 3 - 8 - 6
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 4 - 1 - 5 - 2 - 7 - 8 - 3 - 6
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 4 - 8 - 5 - 6 - 7 - 1 - 3 - 2
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 4 - 8 - 5 - 6 - 7 - 3 - 1 - 2
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 5 - 3 - 1 - 4 - 2 - 7 - 8 - 6
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 5 - 3 - 1 - 4 - 2 - 8 - 7 - 6
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 6 - 4 - 3 - 5 - 8 - 1 - 7 - 2
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 6 - 4 - 3 - 5 - 8 - 7 - 1 - 2
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 8 - 6 - 2 - 4 - 1 - 5 - 7 - 3
Bravo, vous avez gagné !
Les chiffres à trouver sont 9 - 8 - 6 - 2 - 4 - 1 - 7 - 5 - 3
Le nombre de solutions que l'ordinateur a trouvé est 128 !
```



Exercice 10: factorielle

Calculer une factorielle est le processus qui consiste à multiplier un nombre entier par tous les nombres entiers positifs inférieurs à celui-ci.

Exemples :

factorielle de 2 (noté 2 !) = $1 * 2 = 2$

factorielle de 3 (noté 3 !) = $1 * 2 * 3 = 6$

factorielle de 7 (noté 7 !) = $1 * 2 * 3 * 4 * 5 * 6 * 7 = 5040$

- Ecrire une fonction « factorielle » qui reçoit un nombre et retourne la factorielle de ce nombre.
- Ecrire un programme qui affiche toutes les factorielles des nombres inférieurs au nombre saisi.

Exercice 11 : Pyramide

- Ecrire un programme qui affiche un triangle d'étoiles en fonction d'une hauteur donnée.

```
*
**
***
****
*****
*****
*****
*****
```

- Ecrire un programme qui affiche une ligne oblique en fonction d'une hauteur donnée.

```
      *
     *
    *
   *
  *
 *
*
```

- Ecrire un programme qui affiche une pyramide en fonction d'une hauteur donnée.

```
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
```