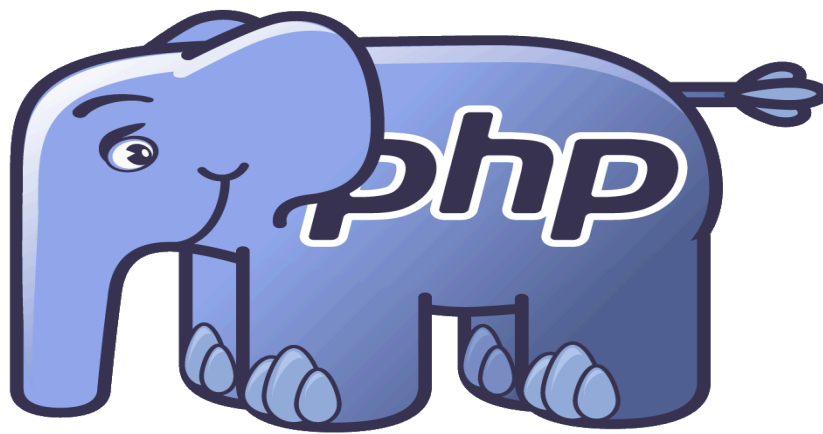




Université Abdelmalek Essaadi
Ecole Nationale des Sciences Appliquées Al-Hoceima



Compte Rendu TP : PHP



Première Année : TDIA-1

Développement web

Réalisée par : KALACH Siham

Encadré par : Pr. Mohamed CHERRADI

Année Universitaire : 2023/2024

Table des matières

I.	EXERCICE 1	-----	2
II.	EXERCICE 2	-----	3
III.	EXERCICE 3	-----	3
IV.	EXERCICE 4	-----	4

Remarque : Pour chaque exercice , il est accompagné par deux parties : html (contient à l'intérieur le code php) , css (pour styler l'affichage de programme).

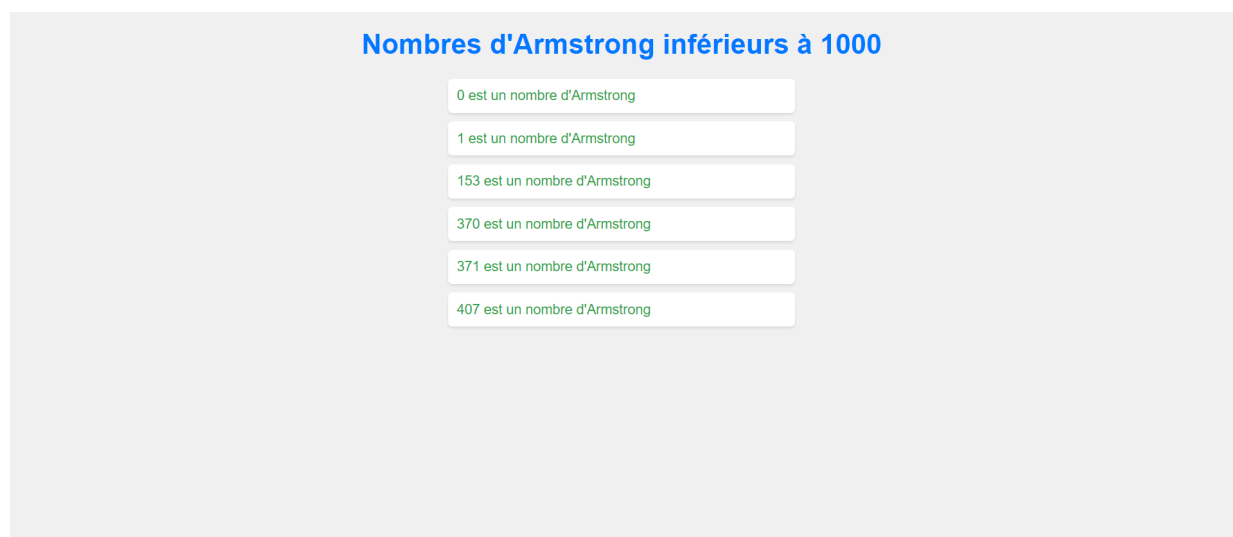
EXERCICE 1 :

On appelle nombres d'Armstrong les nombres entiers positifs tels que la somme des cubes de leurs chiffres est égale au nombre lui-même. Exemple : 153 est un nombre d'Armstrong.

En effet : $1*1*1+5*5*5+3*3*3 = 153$

Écrire un programme PHP qui affiche tous les nombres d'Armstrong inférieurs à 1000 sous forme d'une liste à puce.

Resultat :



Le programme parcourt tous les nombres de 0 à 999 inclus à l'aide d'une boucle for. Pour chaque nombre, il convertit celui-ci en une chaîne de caractères pour pouvoir

accéder à chaque chiffre individuellement. Ensuite, il calcule la somme des cubes de ces chiffres à l'aide d'une autre boucle `for`. Si cette somme est égale au nombre initial, alors ce nombre est considéré comme un nombre d'Armstrong et est ajouté à la liste affichée dans la page HTML.

EXERCICE 2 :

Ecrire un programme PHP qui effectue une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée de deux nombres pairs suivis d'un nombre impair.

Resultat :

10 76 69

EXERCICE 3 :

Ecrire un programme PHP qui permet de générer aléatoirement un nombre de trois chiffres N. Ensuite il effectue des tirages aléatoires jusqu'à réobtenir le même nombre N puis il affiche le nombre d'essais réalisés.

Réalisez ce script d'abord avec l'instruction **while** puis avec l'instruction **for**.

Resultat :

En utilisant while	En utilisant for
Résultat	Résultat
Le nombre aléatoire est : 891	Le nombre aléatoire est : 678
Nombre d'essais : 170	Nombre d'essais : 2237

Le programme génère deux nombres aléatoires à trois chiffres différents en utilisant deux

méthodes différentes : une avec une boucle **while** et l'autre avec une boucle **for**. Pour chaque méthode, il génère un nombre aléatoire à l'aide de la fonction **genererNombreAleatoire()**, qui retourne un nombre aléatoire compris entre 100 et 999 (trois chiffres). Ensuite, il utilise une boucle infinie pour générer des nombres aléatoires jusqu'à ce qu'il trouve un nombre identique à celui généré initialement. À chaque itération, il incrémente un compteur pour suivre le nombre d'essais nécessaires pour trouver le nombre recherché. Une fois que le nombre recherché est trouvé, le programme affiche le nombre aléatoire et le nombre d'essais effectués.

Les résultats sont affichés dans deux sections distinctes sur la page HTML, chacune correspondant à une méthode de génération différente (utilisant **while** ou **for**). Chaque résultat est encapsulé dans une balise `<div>` avec une classe CSS "resultat" pour appliquer un style spécifique.

EXERCICE 4 :

Ecrire une application web qui permet de résoudre les équations du deuxième degré en PHP. L'utilisateur aura la possibilité de saisir les données nécessaires via un formulaire et l'application affiche la solution après la soumission de celui-ci.

Resultat :

Résolution d'équations du second degré

Coefficient a :
 Coefficient b :
 Coefficient c :

Resultat :
 Les solutions sont complexes : $x_1 = 0 + 1i$ et $x_2 = 0 - 1i$

Résolution d'équations du second degré

Coefficient a :
 Coefficient b :
 Coefficient c :

Resultat :
 La solution est réelle et unique : $x = 1$

Résolution d'équations du second degré

Coefficient a :
 Coefficient b :
 Coefficient c :

Resultat :
 Les solutions sont réelles et distinctes : $x_1 = 3$ et $x_2 = 2$

Lorsque l'utilisateur soumet le formulaire, le programme récupère les coefficients saisis, les

utilise pour résoudre l'équation du second degré à l'aide d'une fonction PHP nommée `resoudreEquation()`, puis affiche le résultat sur la page web.

La fonction `resoudreEquation()` calcule le discriminant Δ de l'équation, puis détermine le type de solution en fonction de la valeur de ce discriminant. Si le discriminant est positif, le programme affiche les deux solutions réelles et distinctes. Si le discriminant est nul, il affiche une solution réelle unique. Si le discriminant est négatif, il affiche les deux solutions complexes.