

Ecole Nationale des Sciences Appliquées Al-Hoceima 1ere année cycle ingénieur La transformation digitale et intelligence artificielle

# Le rapport du Tp 3



Réalisé par : Annee universitaire:

- Maach Zineb 2023-2024

# -l'execution d'un fichier PHP:

- Afin d'executer un code PHP, on doit suivre les etapes suivantes:

#### Activer Apache dans XAMPP :

Assurez-vous que le module Apache est activé en XAMPP, en cliquant sur le bouton "Start à côté de "Apache".

#### • Créer un fichier PHP dans le dossier htdocs :

- -Accédez au dossier où XAMPP est installé. Par défaut, il es installé dans "C:\xampp" sous Windows.
- -Dans le dossier XAMPP, ouvrez le dossier "htdocs". C'est l'endroit où vous devez placer vos fichiers PHP pour qu'ils soient accessibles via votre serveur web local. Renommez le fichier avec l'extension ".php", par exemple "monfichier.php".

## • Accéder au fichier via le navigateur :

- -Ouvrez votre navigateur web
- -Tapez "localhost" dans la barre d'adresse du navigateur et appuyez sur Entrée. Cela vous dirigera vers la page d'accueil de XAMPP.
- -Pour accéder à votre fichier PHP, ajoutez le nom de votre fichier PHP à l'URL. Par exemple, si vous avez nommé votre fichier "monfichier.php", tapez
- "localhost/monfichier.php" dans la barre d'adresse du navigateur et appuyez sur Entrée.
- -Vous devriez voir le résultat de l'exécution de votre code PHP dans le navigateur.

-En suivant ces étapes, vous pourrez exécuter votre code PHP avec XAMPP sur votre serveur web local.

# Exercices 01:

-Le but: Ecriture d'un programme PHP qui affiche tous les nombres d'Armstrong inférieurs à 1000 sous forme d'une liste à puce.

-Expliquons le principe de cet exercice:

#### DÉBUT

```
Étape 1 → Prendre la variable $i
```

Étape 2 → Séparer tous les chiffres de \$i.

Étape 3 → Trouver la valeur au cube de chaque chiffre.

Étape  $4 \rightarrow$  Ajouter toutes les valeurs au cube ensemble.

Étape  $5 \rightarrow$  Enregistrer la sortie dans la variable \$s.

Étape 6 → Si \$s est égale à \$i, afficher Nombre d'Armstrong.

Étape 8 → Si \$s n'est pas égale à \$i, l'afficher comme etant un Nombre d'Armstrong.

FIN

-En appliquant cet algortihme, et on obtient le script suivant:

```
Ex1.php
     <?php
     echo "";
     for($i=0; $i<=1000; $i++){
      $s=0;
      $temp=$i;
      while($temp >0){
      $r= $temp % 10;
      s += pow(sr,3);
      $temp /= 10;
10
11
     if($s == $i)
       echo " $i ";
12
13
     echo "";
15
      ?>
```

-Apres l'avoir ouvrir, on obtient:



- 0
- 1
- 153
- 370
- 371
- 407

# Exercice 02:

-Le but: Ecrire un programme PHP qui effectue une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée de deux nombres pair suivis d'un nombre impair.

- Expliquons le principe de cet exercice:

# DÉBUT

Étape 1 → Définir une fonction pour vérifier si un nombre est pair.

Étape 2 → Générer un nombre aléatoire entre 100 et 1000.

Étape 3 → Extraire les chiffres du nombre généré.

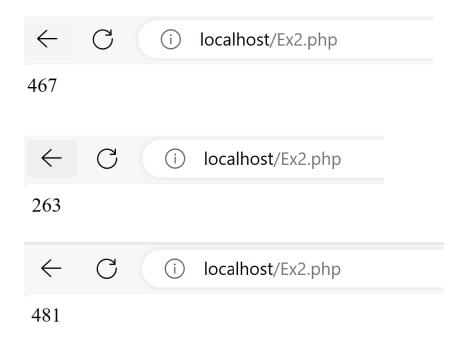
Étape 4 → Si le premier chiffre est pair, le deuxième chiffre est pair et le troisième chiffre est impair : Afficher le nombre généré.

Étape  $5 \rightarrow$  Sinon, revenir à l'étape 2.

#### FIN

-En appliquant cet algortihme, et on obtient le script suivant:

-Le resultat: A chaque fois , il y'a une generation d'un nombre qui verifie les conditions .



# Exercice 04:

- -Le but: Ecriture d'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'essaie s réalisés.
  - -Expliquons le principe de cet exercice:

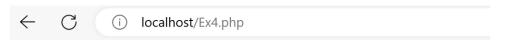
# DÉBUT

- Étape 1 → Générer un nombre aléatoire N entre 100 et 1000.
- Étape 2 → Afficher le nombre généré.
- Étape 3 → Initialiser un compteur de tirages à 0.
- Étape 4 → Tant que le nombre aléatoire M n'est pas égal à N , on génére un nouveau nombre aléatoire M entre 100 et 1000 et en Incrémentant le compteur de tirages, jusqu'a on obtient a nouveau N.
  - Étape 5 → Afficher le nombre de tirages effectués pour obtenir à nouveau N.

#### FIN

- -En appliquant cet alogorithme, on obtient:
  - a)- Avec l'instruction 'while':

-Le resultat: A chaque fois Un nombre est genere, il s'effectue plusieurs de tirages afin qu'on obtient a nouveau le meme nombre.



le nombre c'est :281

On a effectue 1639de tirages pour obtenir une autre fois le nbr:281

# b)- Avec l'instruction 'for':



le nombre c'est :529

On a effectue 798de tirages pour obtenir une autre fois le nbr:529

# Exercice 6:

-Le but: Ecrire une application web qui permet de résoudre les équations du deuxième dégrée 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.

-Expliquons le principe de cet exercice:

## DÉBUT

Étape 1 → Si le formulaire est soumis, on récupére les coefficients a, b et c.

Étape 2 → Calculer le discriminant D.

Étape 6 → Afficher les solutions de l'équation.

Étape 7 → Sinon, afficher le formulaire pour saisir les coefficients a, b et c.

#### FIN

-En appliquant cet alogorithme, on obtient:

-Le resultat: On obteint au 1er lieu, une formulaire a remplire par les coefficients de l'equation , puis s'effectue sa resolution.

$\leftarrow$	C i localhost/Ex5.php
Equa	tion de 2eme degre
Coeff	a:
Coeff	b:
Coeff	c:
Solution	on
← →	O (i) localhost/Ex5.php
Equatio	on de 2eme degre
Coeff a : [	1
Coeff b : [	0
Coeff c : [	-16
Solution	
← 0	i localhost/Ex5.php

# **Equation de 2eme degre**

Les solutions de l'equation  $1 \times 2 + 0x + -16$  sont les suivantes:  $x_1 = 4$  et  $x_2 = -4$