

Université Abdelmalek Essaadi
École Nationale des Sciences Appliquées Al-Hoceima



Rapport: TP N° 3-PHP

**Première Année Transformation digitale &
Intelligence artificielle**

Réalisé par :
YOUSSEF EL KAHLAOUI

Encadré par :
Pr. Mohamed CHERRADI

Table des matières

Rapport: TP N° 3-PHP	1
I. Exercice 1:	2
1. Description :	2
2. Reponse :	2
II. Exercice 2:	3
1. Description :	3
2. Reponse:	3
III. Exercice 3:	3
1. Description :	3
2. Reponse:	3
IV. Exercice 4:	4
1. Description :	4
2. Reponse:	5

I. Exercice 1:

1. Description :

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.

2. Reponse :

```
<?php
echo "Nombres d'Armstrong inférieurs à 1000 :\n";
for ($i = 1; $i < 1000; $i++) {
    $num = $i;
    $sum = 0;
    while ($num != 0) {
        $digit = $num % 10;
        $sum += $digit ** 3;
        $num = (int)($num / 10);
    }
    if ($sum == $i) {
        echo "- $i\n";
    }
}
?>
```

II. Exercice 2:

1. Description :

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.

2. Reponse:

```
<?php
echo "Suite de tirages aléatoires jusqu'à obtenir 2 nombres
pairs suivis d'un nombre impair :\n";
do {
    $random1 = rand(1, 100);
    $random2 = rand(1, 100);
    $random3 = rand(1, 100);
} while ($random1 % 2 == 0 && $random2 % 2 == 0 &&
$random3 % 2 == 0);

echo "Suite obtenue : $random1, $random2, $random3\n";
?>
```

III. Exercice 3:

1. Description :

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'essaies réalisés.

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

2. Reponse:

avec while :

```
<?php
```

```

$N = rand(100, 999);
$es = 1;
$supp = rand(100, 999); #supp = supposition
while ($supp != $N) {
    $supp = rand(100, 999);
    $es++;
}
echo "Nombre de trois chiffres généré aléatoirement : $N\n";
echo "Nombre d'essais réalisés : $es\n";
?>

```

avec for :

```

<?php
$N = rand(100, 999);
$es = 0;
for (;;) {
    $supp = rand(100, 999);
    $es++;
    if ($supp == $N) {
        echo "Nombre de trois chiffres généré aléatoirement :
$N\n";
        echo "Nombre d'essais réalisés : $es\n";
        break;
    }
}
?>

```

IV. Exercice 4:

1. Description :

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.

2. Reponse:

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $a = $_POST["a"];
    $b = $_POST["b"];
    $c = $_POST["c"];

    $delta = $b * $b - 4 * $a * $c;
    if ($delta > 0) {
        $x1 = (-$b + sqrt($delta)) / (2 * $a);
        $x2 = (-$b - sqrt($delta)) / (2 * $a);
        echo "Les solutions de l'équation sont : x1 = $x1 et x2 = $x2";
    } elseif ($delta == 0) {
        $x = -$b / (2 * $a);
        echo "L'équation a une solution double : x = $x";
    } else {
        echo "L'équation n'a pas de solution réelle.";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Résolution d'équations du deuxième degré</title>
</head>
<body>
    <h2>Entrez les coefficients de l'équation  $ax^2 + bx + c = 0$ </h2>
    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
        a: <input type="number" name="a"><br>
        b: <input type="number" name="b"><br>
        c: <input type="number" name="c"><br>
        <input type="submit">
```

```
</form>  
</body>  
</html>
```