

# Fonctions

## TP 04 du Module 02 – Les bases du langage PHP

### Proposition de solution

- Exercice 1

tp4q1f.php

```
<?php
function nbEssaisPour($cible) {
    $nbTentatives = 0;
    while (rand(0, 999) != $cible)
        $nbTentatives++;
    return $nbTentatives;
}
```

tp4q1.php

```
<?php
require_once 'tp4q1f.php';

echo 'Nombre de tentatives pour tomber aléatoirement sur un nombre à trois '
. 'chiffres : ' . nbEssaisPour(666);
```

La fonction est définie dans un fichier à part. Le script, ayant besoin de connaître la définition de cette fonction, inclut son code grâce à l'instruction `require_once`.

#### Exécution

Nombre de tentatives pour tomber aléatoirement sur un nombre à trois chiffres : 795

- Exercice 2

tp4q2f.php

```
<?php
function ordonnerDesc(&$a, &$b) {
    if($a < $b) {
        $t = $a;
        $a = $b;
        $b = $t;
    }
}
```

tp4q2.php

```
<?php
require_once 'tp4q2f.php';

$a = rand(1, 10);
$b = rand(1, 10);

echo 'Initialement : ' . $a . ' ; ' . $b . '<br>';
ordonnerDesc($a, $b);
echo 'Dans l\'ordre décroissant : ' . $a . ' ; ' . $b . '<br>';
```

Les paramètres doivent être passés par référence avec le caractère `&` sinon les modifications effectuées sur ces paramètres ne modifieraient pas les valeurs des variables du programme principal.

**Exécution**

Initialement : 6 ; 8

Dans l'ordre décroissant : 8 ; 6

## • Exercice 3

tp4q3f.php

```
<?php
require_once 'tp4q2f.php';
```

```
function pgcd($a, $b) {
    ordonnerDesc($a, $b);
    do {
        $r = $a % $b;
        $a = $b;
        $b = $r;
    } while ($r !== 0);
    return $a;
}
```

tp4q3.php

```
<?php
require_once 'tp4q3f.php';
```

```
$a = rand(1, 10000);
$b = rand(1, 10000);
```

```
echo 'Le plus grand diviseur commun de ' . $a . ' et ' . $b . ' est ' .
    pgcd($a, $b) . '<br>';
```

La fonction définie à l'exercice précédent est utilisée pour s'assurer que la valeur de la variable **\$a** soit plus grande ou égale à celle de **\$b** avant de commencer la boucle.

**Exécution**

Le plus grand diviseur commun de 7688 et 3255 est 31

## • Exercice 4

tp4q4f.php

```
<?php
function trianglePascal($taille) {
    $p[] = [1];
    for ($i = 1; $i < $taille; $i++) {
        $ligne[0] = 1;
        for ($j = 1; $j < $i; $j++)
            $ligne[$j] = $p[$i - 1][$j - 1] + $p[$i - 1][$j];
        $ligne[$i] = 1;
        $p[] = $ligne;
    }
    return $p;
}
```

```
function afficherTrianglePascal($taille) {
    $pascal = trianglePascal($taille);
    $nbChiffres = floor(log10($pascal[$taille - 1][$taille / 2])) + 2;
    $format = '%' . $nbChiffres . 'd';
    echo '<pre>';
    foreach ($pascal as $ligne) {
        foreach ($ligne as $coef) {
            printf($format, $coef);
        }
    }
}
```

```

        echo '<br>';
    }
    echo '</pre>';
}
tp4q4.php
<?php
require_once 'tp4q4f.php';
afficherTrianglePascal(rand(1,20));

```

La première ligne du triangle de Pascal est créée à part (`$p[] = [1];`).

Voici comment sont produites les lignes suivantes :

- La première valeur est toujours 1 (`$ligne[0] = 1;`);
- La dernière valeur est toujours 1 aussi (`$ligne[$i] = 1;`);
- Pour les autres valeurs, il faut additionner deux valeurs de la ligne précédente (`$ligne[$j] = $p[$i - 1][$j - 1] + $p[$i - 1][$j];`).

Pour la méthode affichant le triangle de Pascal, la principale difficulté est d'aligner les coefficients en colonnes. Les premières valeurs ne sont que sur un seul chiffre, mais dès la sixième ligne, il y a des coefficients sur deux chiffres...

La valeur la plus grande d'un triangle de Pascal est située au milieu de la dernière ligne. Pour connaître le nombre de chiffres dans ce nombre, il est possible de faire appel au logarithme à base 10 (`log10()`). La fonction `floor()` effectue un arrondi à l'entier inférieur.

### Exécution

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1

```