Exercices

Ces exercices vous donneront des exemples d'algorithmes répondant à une problématiques données. L'ensemble de ces algorithmes peut former une base de révision algorithmique car ce document couvre les éléments nécessaires pour construire la plupart des algorithmes procéduriaux existant.

Problèmes

Vous trouverez la liste des problèmes et une de leurs solutions algorithmiques ci-dessous.

1. Faire l'addition de 2 nombres

Problème : Ecrivez une fonction qui prend 2 nombres flottants et qui retourne l'addition des 2 nombres

Réponse à l'exercice :

```
function additionner(int $a, int $b): int
{
    return $a + $b;
}
```

Pour tester la function :

```
var_dump(additionner(3, 4));
```

2. Vérifier si un utilisateur est majeur

Problème : Ecrivez une fonction qui prend un `age` en paramètre et retourne `vrai` si l'utilisateur est majeur et `faux` sinon

Réponse à l'exercice :

```
function verifierMajeur(int $age): bool
{
    if ($age >= 18)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Pour tester la function :

```
var_dump(verifierMajeur(23));
var_dump(verifierMajeur(7));
```

4. Vérifier si un utilisateur est mineur

Problème : Ecrivez une fonction qui prend un `age` en paramètre et retourne `vrai` si l'utilisateur est mineur et `faux` sinon

Réponse à l'exercice :

```
function verifierMineur(int $age): bool
{
    if ($age < 18)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

Pour tester la function :

```
var_dump(verifierMineur(23));
var_dump(verifierMineur(7));
```

5. Echanger les 2 premiers éléments d'un tableau

Problème : Ecrivez une fonction qui prend un `tableau` en paramètre et retourne un `tableau` avec les 2 premiers éléments échangés

Réponse à l'exercice :

```
function copierEtEchanger(array $tab): array
{
    $copie = $tab;
    $copie[0] = $tab[1];
    $copie[1] = $tab[0];
```

```
    return $copie;
}
```

Pour tester la function :

```
var_dump(copierEtEchanger([23, 1, 45, 3, 22, 3]));
```

6. Compter le nombre d'éléments dans un tableau

Problème : Ecrivez une fonction qui compter le nombre d'éléments dans un tableau. Comparez le résultat avec le resultat de la fonction `count`

Réponse à l'exercice :

```
function compter(array $tab): int
{
    $compte = 0;
    foreach ($tab as $element)
    {
        $compte = $compte + 1;
    }
    return $compte;
}
```

Pour tester la function :

```
var_dump(compter([23, 1, 45, 3, 22, 3]));
```

7. Faire la somme des éléments d'un tableau

Problème : Ecrire une fonction qui prend un `tableau` en paramètre et qui retourne la somme des éléments du tableau

Réponse à l'exercice :

```
function sommer(array $tab): float
{
    $somme = 0;
    foreach ($tab as $nombre)
    {
        $somme = $somme + $nombre;
```

```
    }
    return $somme;
}
```

Pour tester la function :

```
var_dump(sommer([23, 1, 45, 3, 22, 3]));
```

8. Trouver l'élément maximum d'un tableau

Problème : Ecrire une fonction qui prend un `tableau` en paramètre et retourne l'élément maximum

Réponse à l'exercice :

```
function trouverMaximum(array $tab): float
{
    $retenu = $tab[0];
    foreach ($tab as $nombre)
    {
        if ($retenu < $nombre)
        {
            $retenu = $nombre;
        }
    }
    return $retenu;
}
```

Pour tester la function :

```
var_dump(trouverMaximum([23, 1, 45, 3, 22, 3]));
```

9. Trouver l'élément minimum d'un tableau

Problème : Ecrire une fonction qui prend un `tableau` en paramètre et retourne l'élément minimum

Réponse à l'exercice :

```
function trouverMinimum(array $tab): float
{
    $retenu = $tab[0];
    foreach ($tab as $nombre)
    {
```

```
        if ($retenu > $nombre)
        {
            $retenu = $nombre;
        }
    }
    return $retenu;
}
```

Pour tester la function :

```
var_dump(trouverMinimum([23, 1, 45, 3, 22, 3]));
```

10. Trouver le premier élément supérieur à un nombre

Problème : Ecrire une fonction qui prend un `tableau` et un nombre floattant en paramètre et retourne le premier élément supérieur à ce nombre s'il y en a, sinon null. Vous pourrez utiliser le type `?float` à la place du type `float`

Réponse à l'exercice :

```
function trouverPremierElement(array $tab, float $seuil): ?float
{
    foreach ($tab as $nombre)
    {
        if ($nombre > $seuil)
        {
            return $nombre;
        }
    }
    return null;
}
```

Pour tester la function :

```
var_dump(trouverPremierElement([23, 1, 45, 3, 22, 3], 30));
var_dump(trouverPremierElement([23, 1, 45, 3, 22, 3], 50));
```

11. Copier un tableau

Problème : Ecrire une fonction qui prend un `tableau` et qui le copie dans un autre tableau `autre`, vous devez utiliser la fonction `array_push`

Réponse à l'exercice :

```php
function copier(array $tab): array
{
    $nouveau = [];
    foreach ($tab as $element)
    {
        array_push($nouveau, $element);
    }
    return $nouveau;
}
```

Pour tester la function :

```php
var_dump(copier([23, 1, 45, 3, 22, 3]));
```

12. Copier les premiers éléments d'un tableau dont la somme fait au moins un nombre

Problème : Ecrire une fonction qui prend un `tableau` et un nombre entier et qui copie les premiers éléments d'un tableau dont la somme fait au moins ce nombre

Réponse à l'exercice :

```php
function copierPourSomme(array $tab, float $somme): array
{
    $sommeEnCours = 0;
    $elements = [];
    foreach ($tab as $nombre)
    {
        array_push($elements, $nombre);
        $sommeEnCours = $sommeEnCours + $nombre;

        if ($sommeEnCours >= $somme)
        {
            return $elements;
        }
    }
    return [];
}
```

Pour tester la function :

```php
var_dump(copierPourSomme([23, 1, 45, 3, 22, 3], 100.0));
var_dump(copierPourSomme([23, 1, 45, 3, 22, 3], 900.0));
```

13. Copier les premiers éléments d'un tableau dont la somme fait au moins un nombre

Problème : Ecrire une fonction qui prend un `tableau` et un nombre entier et qui copie les premiers éléments d'un tableau dont la somme fait au moins ce nombre, vous devez utiliser la fonction `array_sum`

Réponse à l'exercice :

```
function copierPourSomme(array $tab, float $somme): array
{
    $elements = [];
    foreach ($tab as $nombre)
    {
        array_push($elements, $nombre);
        $sommeEnCours = array_sum($elements);

        if ($sommeEnCours >= $somme)
        {
            return $elements;
        }
    }
    return [];
}
```

Pour tester la function :

```
var_dump(copierPourSomme([23, 1, 45, 3, 22, 3], 100.0));
var_dump(copierPourSomme([23, 1, 45, 3, 22, 3], 900.0));
```

14. Mapper des données

Problème : Ecrire une fonction qui prend un tableau de tableaux associatifs (produits) et qui renvoie un tableau de tableaux associatifs (produits) dont chaque produit contient le prix total (prix unitaire multiplié par la quantité)

Réponse à l'exercice :

```
function mapper(array $tab): array
{
    $resultat = [];
    foreach($tab as $produit)
    {
        $produit['total'] = $produit['prix'] * $produit['quantite'];
        array_push($resultat, $produit);
    }
```

```
    return $resultat;
}
```

Pour tester la function :

```
var_dump(mapper(
    [
        [
            'prix' => 9.8,
            'quantite' => 2
        ],
        [
            'prix' => 3.2,
            'quantite' => 3
        ],
        [
            'prix' => 10.9,
            'quantite' => 1
        ]
    ]
));
```

15. Mapper des données

Problème : Ecrire une fonction qui prend un tableau de chaines de caractères et qui renvoie un tableau de chaines de caractères dont chaque chaine de caractères a été tronquée au 5ème caractère

Réponse à l'exercice :

```
function tronquer(array $tab): array
{
    $resultat = [];
    foreach($tab as $texte)
    {
        $texteTronque = substr($texte, 0, 5);
        array_push($resultat, $texteTronque);
    }

    return $resultat;
}
```

Pour tester la function :

```
```

```
var_dump(mapper(
    [
        'La tomate trop bonne', 'Je dis oui, a la tomate', 'Le retour de
Madamde Tomate', 'La Philosophie de la tomate'
    ]
));
```

16. Mapper des données

Problème : Ecrire une fonction qui prend un tableau de tableaux associatifs d'utilisateurs et qui renvoie un tableau de tableaux associatifs d'utilisateurs avec seulement le nom, prenom et age

Réponse à l'exercice :

```
function mapper(array $tab): array
{
    $resultat = [];
    foreach($tab as $utilisateur)
    {
        $nouvelUtilisateur = [
            'nom' => $utilisateur['nom'],
            'prenom' => $utilisateur['prenom'],
            'age' => $utilisateur['age'],
        ];
        array_push($resultat, $nouvelUtilisateur);
    }

    return $resultat;
}
```

Pour tester la function :

```
var_dump(mapper(
    [
        [
            'nom' => 'Aimarre',
            'prenom' => 'Jean',
            'age' => 27,
            'mot_de_passe' => '08-03-1982'
        ],
        [
            'nom' => 'Riendetoit',
            'prenom' => 'Johnatan',
            'age' => 45,
            'mot_de_passe' => '08-03-1982'
        ],
```

```
        [
            'nom' => 'Tu',
            'prenom' => 'Candy',
            'age' => 26,
            'mot_de_passe' => '08-03-1982'
        ],
        [
            'nom' => 'Thare',
            'prenom' => 'Guy',
            'age' => 41,
            'mot_de_passe' => '08-03-1982'
        ]
    ]
));
```

17. Mapper des données

Problème : Ecrire une fonction qui prend un tableau de tableaux associatifs (produits) et qui renvoie un tableau de tableaux associatifs (produits) dont le prix a été diminué de 20 %

Réponse à l'exercice :

```
function mapper(array $tab): array
{
    $resultat = [];
    foreach($tab as $produit)
    {
        $produit['prix'] = $produit['prix'] * 0.80;
        array_push($resultat, $produit);
    }

    return $resultat;
}
```

Pour tester la function :

```
var_dump(mapper(
    [
        [
            'prix' => 9.8,
            'quantite' => 2
        ],
        [
            'prix' => 3.2,
            'quantite' => 3
        ],
```

```
            [
                'prix' => 10.9,
                'quantite' => 1
            ]
        ]
    ));
```

18. Filtrer des données

Problème : Ecrire une fonction qui prend un tableau de tableaux associatifs (commandes) et qui renvoie un tableau de tableaux associatifs (commandes) à traiter, c'est-a-dire dont la date d'expédition est dans le futur

Réponse à l'exercice :

```
function filtrer(array $tab): array
{
    $commandeAPreparer = [];
    $dateCourante = date('Y-m-d');
    foreach ($tab as $commande)
    {
        if ($commande['expedition'] > $dateCourante)
        {
            array_push($commandeAPreparer, $commande);
        }
    }
    return $commandeAPreparer;
}
```

Pour tester la function :

```
var_dump(mapper(
    [
        [
            'reference' => '0383954',
            'expedition' => '2021-05-16'
        ],
        [
            'reference' => '676985',
            'expedition' => '2021-04-26'
        ],
        [
            'reference' => '987695',
            'expedition' => '2021-07-20'
        ]
    ]
));
```

19. Filtrer des données

Problème : Ecrire une fonction qui prend un tableau de tableaux associatifs (produits) et un nombre entier N et qui renvoie un tableau de tableaux associatifs (produits) des N produits les moins cher

Réponse à l'exercice :

```php
function comparerDeuxProduitParPrix(array $produit1, array $produit2): int
{
    return $produit1['prix'] <=> $produit2['prix'];
}

function filtrer(array $tab, int $n): array
{
    usort($tab, 'comparerDeuxProduitParPrix');
    $produitsPasCher = [];
    $compte = 0;
    foreach ($tab as $produit)
    {
        if ($compte == $n)
        {
            return $produitsPasCher;
        }
        else
        {
            array_push($produitsPasCher, $produit);
            $compte = $compte + 1;
        }
    }

    return $produitsPasCher;
}
```

Pour tester la function :

```php
var_dump(filtrer(
    [
        [
            'prix' => 23.2,
            'nom' => 'Nourriture pour papier toilette'
        ],
        [
            'prix' => 9.8,
            'nom' => 'Papier toilette'
        ],
        [
```

```
            'prix' => 340.5,
            'nom' => 'Collier amour pour papier toilette'
        ],
        [
            'prix' => 3.2,
            'nom' => 'Maison pour papier toilette'
        ],
        [
            'prix' => 10.9,
            'nom' => 'Savon pour papier toilette'
        ]
    ]
));
```

20. Bonus

Problème : On cherche à faire une application web mobile pour apprendre à voler un vélo qui est cadenassé avec un cadenas à code (il y a des molettes à tourner). Ecrire une fonction qui prend en paramètre un nombre entier (représentant le nombre de molettes à tourner) et qui retourne un tableau de chaines de caractères contenant toutes les combinaisons de codes possibles.

Réponse à l'exercice :

```
function trouverCombinaisons(int $nombreMolettes): array
{
    $combinaisonMaximum = 10 ** $nombreMolettes -1;
    $combinaisons = [];
    for ($i = 0; $i <= $combinaisonMaximum; ++$i)
    {
        $possibilite = str_pad(strval($i), $nombreMolettes, '0',
STR_PAD_LEFT);
        array_push($combinaisons, $possibilite);
    }
    return $combinaisons;
}
```

Pour tester la function :

```
var_dump(trouverCombinaisons(4));
```