

M-10 Exercices d'algorithmie PHP

Conditions :

1. Le Juste Prix

Écrivez un algorithme qui demande à l'utilisateur de deviner un nombre secret (entre 1 et 100).

- Si le nombre est trop grand, affichez "C'est moins !".
- S'il est trop petit, affichez "C'est plus !".
- Si c'est le bon nombre, affichez "Bravo !".
- L'utilisateur a maximum 10 tentatives. À la fin, si le nombre n'est pas trouvé, affichez "Perdu, le nombre était X".

2. Validateur de Mot de Passe

Écrivez un algorithme qui vérifie si un mot de passe respecte les conditions suivantes :

- Au moins 8 caractères.
- Contient au moins une majuscule.
- Contient au moins un chiffre.
- **Bonus** : contient au moins un caractère spécial (!@#\$%^&*, etc.).

3. Année Bissextile

Écrivez un algorithme qui détermine si une année est bissextile.

- Une année est bissextile si elle est divisible par 4, sauf si elle est divisible par 100 mais pas par 400.

4. Jour de la Semaine

Demandez à l'utilisateur un nombre entre 1 et 7 et affichez le jour de la semaine correspondant (1 = lundi, 2 = mardi, etc.).

- Si le nombre n'est pas compris entre 1 et 7, affichez "Erreur : nombre invalide".

5. **Calculateur de Coût**

Un service de covoiturage applique les tarifs suivants :

- 1€ par km pour les 10 premiers km.
- 0.8€ par km pour les 20 km suivants.
- 0.5€ par km pour le reste du trajet.

Écrivez un algorithme qui calcule le coût total d'un trajet en fonction de la distance.

- Si la distance est 0 km, affichez "Coût = 0€".

Boucles :

1. **Compte à Rebours**

Affichez un compte à rebours de 10 à 0, puis affichez "Décollage !".

- Variante : affichez les nombres soit sur une seule ligne séparée par des espaces, soit ligne par ligne.

2. **Somme des N Premiers Nombres**

Demandez un nombre N à l'utilisateur et calculez la somme de tous les entiers de 1 à N.

- *Bonus* : proposez aussi la solution avec la formule mathématique $N*(N+1)/2$.

3. **Table de Multiplication**

Demandez un nombre N à l'utilisateur et affichez sa table de multiplication de 1 à 10 sous la forme :

$N \times 1 = \dots$, $N \times 2 = \dots$, etc.

4. **Devinette**

L'ordinateur choisit un nombre aléatoire entre 1 et 10. L'utilisateur a 3 tentatives pour deviner le nombre.

- Après chaque tentative, affichez "Trop grand", "Trop petit" ou "Correct".
- Si l'utilisateur n'a pas trouvé après 3 essais, affichez "Perdu ! Le nombre était X".

5. Afficher des Étoiles

Écrivez un algorithme qui affiche un triangle d'étoiles de N lignes. Exemple pour N=4 :

```
*  
**  
***  
****
```

- *bonus* : afficher un triangle inversé ou une pyramide.

Tableaux et tableaux associatifs :

1. Inversion de Tableau

Inversez les éléments d'un tableau sans utiliser de fonction native comme `reverse()`.

2. Recherche de Valeur

Écrivez un algorithme qui vérifie si un élément est présent dans un tableau.

- S'il est présent, renvoyez son ou ses index (s'il apparaît plusieurs fois).
- Sinon, renvoyez -1.

3. Tableau des Notes

Créez un tableau associatif qui stocke les notes de plusieurs élèves (nom => note).

- Calculez la moyenne des notes.
- Indiquez la meilleure et la plus faible note.

4. Fusion de Tableaux Associatifs

Fusionnez deux tableaux associatifs.

- Si une clé existe dans les deux tableaux, la valeur du second tableau remplace celle du premier.

5. Compteur de Mots

À partir d'une chaîne de caractères, créez un tableau associatif qui compte le nombre d'occurrences de chaque mot.

- Ignorez la casse (majuscule/minuscule).
 - Supprimez la ponctuation pour un vrai comptage.
-

Mini-Programmes :

1. Gestion de Stock

Créez un programme qui gère un inventaire sous forme de tableau associatif (article => quantité). Le programme doit permettre :

- Ajouter un article avec une quantité.
- Mettre à jour la quantité d'un article.
- Supprimer un article.
- Afficher les articles dont le stock est inférieur à 10.

2. Jeu de la Pendu

Développez un mini-jeu du pendu.

- Le programme choisit un mot secret.
- L'utilisateur propose une lettre à chaque tentative.
- Affichez l'état du mot après chaque essai (ex. p__g__mme).
- Le jeu s'arrête quand l'utilisateur trouve le mot ou quand il n'a plus de tentatives.

3. Système d'Authentification

Créez un système d'authentification avec un tableau associatif (utilisateur => mot_de_passe).

- Demandez à l'utilisateur ses identifiants.
- S'ils sont corrects, affichez "Bienvenue X".
- Sinon, affichez "Identifiants incorrects".
- Après 3 échecs, affichez "Compte bloqué".

4. Calculatrice en Ligne de Commande

Construisez une calculatrice qui prend deux nombres et un opérateur (+, -, *, /).

- Affichez le résultat.



- Gérez les erreurs (ex : division par zéro).
- *Bonus* : permettre les calculs en chaîne ($2 + 3 * 4$).