

Évaluation - Algorithmique et Mathématiques

Auteurs : N'jie ZAMON, Joachim MARIE-LUCE

Présentation Générale

Cette évaluation est divisée en deux parties :

- **Partie 1 : Algorithmique** - Concevez et implémentez des algorithmes pour résoudre des problèmes spécifiques.
- **Partie 2 : Mathématiques** - Résolvez des exercices basés sur vos connaissances théoriques en mathématiques.

Chaque partie comporte trois exercices indépendants. Vous êtes libre de choisir l'ordre dans lequel vous les abordez. Le temps imparti est d'une heure pour traiter l'ensemble des exercices.

Remarque : Essayez de répondre à tous les exercices, même partiellement. Chaque tentative sera prise en compte dans l'évaluation.

Partie 1 : Algorithmique

Exercice 1 : Tri d'une liste

1. Écrire un algorithme qui prend en entrée une liste de nombres et qui retourne cette liste triée.
2. Implémentez cet algorithme en Python ou dans un autre langage de votre choix. (l'utilisation des fonctions intégrés de Python tel que "sorted" n'est pas autorisé)

Exercice 2 : Explication d'un algorithme

Lisez le programme suivant écrit en Python et fournissez une explication détaillée de son fonctionnement :

```
def divise(dividende, diviseur):  
    if diviseur == 0:  
        return ("Impossible !")  
    quotient = 0  
    while dividende >= diviseur:  
        dividende -= diviseur  
        quotient += 1  
    return quotient
```

Exercice 3 : Calcul de la factorielle

Écrire un algorithme qui calcule la factorielle d'un nombre entier positif. La factorielle d'un nombre n , notée $n!$, est le produit de tous les entiers positifs de 1 à n .

Exemple : $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$.

Conseils :

- Utilisez une boucle pour effectuer les multiplications successives.
- Initialisez le résultat à 1.

Partie 2 : Mathématiques

Exercice 1 : Équations

Résoudre dans R les équations suivantes :

a) $3x^2 + 2x - 1 = 0$

b) $e^{2x} = 7$

c) $\log_2(x - 1) = 4$

Exercice 2 : Multiplication

Rédigez une fonction Python nommée `multiplication`, prenant comme paramètres deux entiers relatifs a et b , et renvoyant le produit de ces deux nombres. La fonction ne doit pas faire appel au signe multiplication (noté `*`). La fonction `abs()` n'est pas autorisée.

Exemple : `multiplication(3, 4)` doit retourner 12. `multiplication(-3, 4)` doit retourner -12.

Conseils :

- Utilisez une boucle et l'opération d'addition répétée pour calculer le produit.
- Pensez à gérer les cas où a ou b sont négatifs.

Exercice 3 : PGCD

Pour calculer le PGCD de deux nombres entiers relatifs a et b (avec $a \geq b$), nous utilisons l'algorithme d'Euclide, qui s'effectue de la manière suivante :

1. **Initialisation :**

- Soit a et b deux entiers relatifs tels que $a \geq b$.

2. **Division :**

- Divisez a par b pour obtenir :
 - q , le quotient (la partie entière du résultat de la division).
 - r , le reste de cette division.

3. **Répétition :**

- Tant que r n'est pas égal à zéro, remplacez a par b et b par r .
- Répétez l'étape 2 pour les nouvelles valeurs de a et b .

4. **Résultat :**

- Lorsque r devient égal à zéro, le PGCD de a et b est la dernière valeur non nulle de b .

Rédigez une fonction Python nommée `PGCD`, prenant comme paramètres deux entiers relatifs a et b , et qui renvoie le PGCD de ces deux nombres. Le signe modulo (noté `%`) n'est pas autorisé.

Exemple :

`PGCD(18, 24)` doit retourner 6.

`PGCD(42, 56)` doit retourner 14.