
Tests préconditions & postconditions

Corrections

1. Exercice : les concepts de **préconditions** & **postconditions**

Vous devez écrire des préconditions et des postconditions pour une fonction Python donnée.

Consignes :

1. Lisez attentivement la fonction ci-dessous.
2. Écrivez les préconditions que vous jugez nécessaires pour que la fonction fonctionne correctement.
3. Écrivez les postconditions qui doivent être vraies après l'exécution de la fonction.

```
def factorielle(n):  
    """  
    Calcule la factorielle d'un nombre entier positif n.  
  
    :param n: Un entier positif  
    :return: La factorielle de n  
    """  
    if n == 0:  
        return 1  
    else:  
        resultat = 1  
        for i in range(1, n + 1):  
            resultat *= i  
        return resultat
```

Travail demandé :

1. **Préconditions** : Écrivez ce qui doit être vrai avant que la fonction soit appelée.

Par exemple :

- Le paramètre n doit être un entier.
- Le paramètre n doit être un entier positif ou nul.

2. **Postconditions** : Écrivez ce qui doit être vrai après que la fonction a été exécutée. Par exemple :

- La sortie doit être un entier.
- La valeur retournée doit être la factorielle correcte de n .

Corrections

```
def factorielle(n):
    """
    Calcule la factorielle d'un nombre entier positif n.

    Préconditions :
    - n doit être un entier.
    - n doit être supérieur ou égal à 0.

    Postconditions :
    - Le résultat retourné doit être un entier.
    - Le résultat doit être égal à n! (factorielle de n).

    :param n: Un entier positif ou nul
    :return: La factorielle de n
    """

    # Vérification des préconditions
    assert isinstance(n, int), "n doit être un entier."
    assert n >= 0, "n doit être supérieur ou égal à 0."

    if n == 0:
        return 1
    else:
        resultat = 1
        for i in range(1, n + 1):
            resultat *= i

    # Vérification des postconditions
    assert isinstance(resultat, int), "Le résultat doit être un entier."
    assert resultat == fact_verifie(n), "Le résultat doit être la factorielle correcte."

    return resultat

def fact_verifie(n):
    """ Calcule la factorielle de manière récursive pour vérifier les postconditions """
    if n == 0:
        return 1
    else:
        return n * fact_verifie(n - 1)
```

Justification de fact_verifie :

La fonction fact_verifie est une implémentation alternative de la factorielle utilisant la récursion. Elle sert simplement à valider que le résultat produit par la fonction principale est correct, conformément à la définition mathématique de la factorielle.

Comportement en cas d'erreur :

Si on appelle la fonction avec des valeurs qui ne respectent pas les préconditions, des erreurs sont levées :

```
print(factorielle(-1)) # Erreur : n doit être supérieur ou égal à 0
```

```
print(factorielle(3.5)) # Erreur : n doit être un entier
```