

## Modèle de correction - Exercice de programmation

### Correction détaillée de l'exercice - Gestion de fichiers

Voici le modèle de correction pour l'exercice sur la gestion des fichiers. Les fonctions sont expliquées une par une avec des exemples de code. Assurez-vous de comprendre chaque partie du code.

#### 1. Problème 1 : Lire un fichier et afficher son contenu

Fonction : ``lire_fichier(nom_fichier)``

- Cette fonction ouvre le fichier, lit son contenu ligne par ligne et l'affiche. Si le fichier n'existe pas, une exception est gérée.

Code :

```
```python
def lire_fichier(nom_fichier):
    try:
        with open(nom_fichier, 'r') as f:
            for ligne in f:
                print(ligne.strip())
    except FileNotFoundError:
        print(f"Erreur : Le fichier {nom_fichier} n'existe pas.")
```
```

Exemple d'appel :

```
```python
lire_fichier('exemple.txt')
```

...

## 2. Problème 2 : Compter les mots dans un fichier

Fonction : ``compter_mots(nom_fichier)``

- Cette fonction lit le fichier et compte le nombre de mots.

Code :

```
```python
def compter_mots(nom_fichier):
    try:
        with open(nom_fichier, 'r') as f:
            contenu = f.read()
            mots = contenu.split()
            return len(mots)
    except FileNotFoundError:
        print(f"Erreur : Le fichier {nom_fichier} n'existe pas.")
```
```

...

Exemple d'appel :

```
```python
print(compter_mots('exemple.txt'))
```
```

## 3. Problème 3 : Écrire dans un fichier

Fonction : ``ecrire_dans_fichier(nom_fichier, texte)``

- Cette fonction écrit le texte dans un fichier, en écrasant le contenu existant.

Code :

```
```python
def ecrire_dans_fichier(nom_fichier, texte):
    try:
        with open(nom_fichier, 'w') as f:
            f.write(texte)
    except IOError:
        print(f"Erreur : Impossible d'écrire dans le fichier {nom_fichier}.")
```
```

Exemple d'appel :

```
```python
ecrire_dans_fichier('exemple.txt', 'Voici un texte à écrire dans le fichier.')
```
```

#### 4. Problème 4 : Fusionner deux fichiers

Fonction : `fusionner\_fichiers(nom\_fichier1, nom\_fichier2, nom\_fichier\_sortie)`

- Cette fonction fusionne le contenu de deux fichiers dans un troisième fichier.

Code :

```
```python
def fusionner_fichiers(nom_fichier1, nom_fichier2, nom_fichier_sortie):
    try:
        with open(nom_fichier1, 'r') as f1, open(nom_fichier2, 'r') as f2, open(nom_fichier_sortie, 'w') as
f_out:
```

```
f_out.write(f1.read())

f_out.write(f2.read())

except FileNotFoundError:

    print("Erreur : L'un des fichiers n'existe pas.")
'''
```

Exemple d'appel :

```
'''python

fusionner_fichiers('fichier1.txt', 'fichier2.txt', 'fichier_fusionne.txt')

'''
```

## 5. Problème 5 : Ajouter une gestion des erreurs

Chaque fonction doit gérer les erreurs de manière appropriée pour éviter des plantages du programme. Les erreurs courantes à gérer sont les fichiers inexistants, les erreurs d'écriture, et les problèmes de permission.

Exemple :

- Utiliser `try` et `except` pour capturer les exceptions de type `FileNotFoundError` ou `IOError`.