

Nom :

Prénom :

Groupe :

Examen blanc**TOTAL :** /**Question :** _____**Les compagnons du Docteur**

Docteur Who a besoin d'une fonction pour lister et décrire ses compagnons. Créez une fonction `analyser_compagnon` qui prend en paramètre le nom d'un compagnon sous forme de chaînes de caractères. La fonction doit renvoyer la description du compagnon qui se trouve dans un dictionnaire extérieur à la fonction.

Le scan de compagnon

Lorsque le Docteur scan un compagnon avec son tournevis sonique, il peut soit ajouter le compagnon et une description fournie par l'utilisateur au dictionnaire s'il n'existe pas, soit le supprimer du dictionnaire, soit modifier sa description. Ecrivez une fonction `scan_compagnon` qui prend en paramètre le nom du compagnon scanné. La fonction doit ensuite vérifier si le compagnon est déjà présent ou pas et agir en conséquence.

Le registre du TARDIS

Le vaisseau du Docteur, le TARDIS, tient dans ses bases de données un registre de tous ses compagnons dans un fichier nommé `doc_registre.txt`. Chaque ligne du fichier contient le nom d'un compagnon. Ecrivez une fonction `lire_registre` pour lire le fichier et compter le nombre d'occurrences de chaque compagnon. Affichez les résultats sous forme d'un dictionnaire.

L'appel à l'aide

Le Docteur a besoin d'un système pour savoir quand et où l'humanité a besoin de lui. Écrivez une fonction `appel_aide` qui utilise une boucle pour simuler un appel à l'aide. Le programme doit demander à l'utilisateur d'entrer 'oui' ou 'non'. Si l'utilisateur entre 'oui', le programme affiche "Le Docteur arrive !". Si l'utilisateur entre 'non', le programme affiche "L'Humanité est en

Nom :

Prénom :

Groupe :

sécurité pour le moment." Le programme doit continuer à demander jusqu'à ce que l'utilisateur entre 'quit' pour quitter la boucle.

Attention : Le Maître s'en mêle !

Le Maître a décidé de jouer un tour au Docteur. Il a ajouté un faux compagnon appelé Missy dans le registre. Modifiez votre script de la Partie 2 pour ignorer ce compagnon lors du comptage.

Pour cet exercice, vous avez reçu un dossier comprenant un main.py ainsi que des fichiers .txt. Vous devez écrire à l'endroit prévu dans le main.py et utiliser les fichiers prévus. Le main.py contient des tests unitaires qui vous permettront de savoir si vous fournissez bien le travail demandé. Vous devez écrire votre code dans les parties prévues à cet effet.

En fin d'exercice, remettre un dossier compressé, qui contient votre code, portant votre nom suivi de votre prénom.

L'utilisation des clés USB, de téléphone, de montres connectées, la mise en réseaux (wifi ou autres) sont interdits durant l'épreuve. Toute tentative de fraude entraînera automatiquement une cote de 0 pour cette épreuve.

Durée maximale de l'exercice : 2h

Bon travail.