1 Test technique

Ce test technique se divise en deux exercices indépendants. Il n'est pas conçut pour être terminé en 1h mais pour vous laisser le choix dans les questions à traiter. À la fin de chaque exercice il y a une question plus difficile qui demande des compétences allant au-dela de ce qui est normalement attendu d'un developpeur junior. Les questions de l'exercice 1 dépendent les unes des autres. Les questions de l'exercice 2 sont independantes.

Pour lancer la correction automatique il suffit d'utiliser la commande `python test_correction.py`. Vous pouvez lancer la correction autant de fois que vous le souhaitez.

1.1 Exercice 1 : Spatioport

1.1.1 Question 1

Modifier la fonction `is_same_ship(ship1, ship2)` pour qu'elle retourne True si les deux vaisseaux ont les mêmes `name`, `size` et `author` et False sinon.

1.1.2 Question 2

Dans la classe Spaceport modifier la méthode `contains(self, ship)` pour qu'elle retourne True si un vaisseau est dans la liste docked et False sinon

1.1.3 Question 3

Dans la classe Spaceport modifier la méthode `dock(self, ship)` pour qu'elle rajoute un vaisseau dans docked si il reste assez de place dans sa catégorie de taille ou dans la catégorie superieur et lève une exception sinon.

1.1.4 Question 4

Dans la classe Spaceport modifier la méthode `undock(self, ship)` pour qu'elle retire un vaisseau de la liste docked si le vaisseau y est et lève une exception sinon

1.1.5 Question 5

Question avancée: ecrire une fonction qui permet de sauvegarder l'état du spaceport dans un fichier.

note: cette fonction n'à pas de correction. automatique

1.2 Exercice 2 : Questions diverses

1.2.1 Question 1

Modifier la fonction qui `filter_letters(string)` qui prend en entré une string en majuscule et qui retourne une liste de lettres présentes dans string ayant un rang impaire (A, C, E, G, etc...)

exemple : `filter_letters("chaise") -> "caie"`

1.2.2 Question 2

Modifier la fonction `count_letters(string)` qui pernd en entré une string et qui retourne un dictionnaire avec en clef chaque caractère de la string et en valeur le nombre d'occurence de ce caractère.

exemple : `count_letters("lettre") -> {"l": 1, "e": 2, "t": 2, "r": 1}`

1.2.3 Question 3

Modifier la fonction `mask_letters(string, mask)` qui prend en entrée 2 string et retourne une string. La premiere est un texte, la deuxieme (le masque) est une string de même longueur composé de 0 et de 1. pour chaque rang ou le masque est un zéro, la lettre de même rang de la string est ignorée, si le charactere du masque est un 1 le charactere de même rang est ajouté a la string de retour.

exemple : `mask_letters("une longue string", "10011100101010101") -> "u lonu tig"`

1.2.4 Question 4

Modifier la fonction `puissances(start, end)` qui prend deux entier (avec start > end) et qui retourne la liste des carrés parfaits de l'interval entre ces deux entiers inclus.

exemple: `` `puissance(1, 5) -> [1, 4, 9, 16, 25 ] ``

1.2.5 Question 5

Question avancée: Sans modifier le code à l'interieur de la fonction `return_int` faies en sorte que cette fonction retourne une string à la place d'un entier.