

LINFO1002-2020-A-7

Question 1 : REPLACE

Tout d'abord, à partir des évaluations faites sur cette fonction (`replace()`), la première chose qui manquait était une description claire de ce qui était censé être fait. Nous avons donc décidé de le rendre plus clair et compréhensible.

Deuxièmement, des spécifications et des exemples ont été ajoutées pour diriger explicitement l'étudiant vers une implémentation correcte, ainsi rendant les choses plus claires et dans le format correct, comme nous l'avons demandé.

Troisièmement, nous avons ajouté plus de tests afin que lorsque l'utilisateur a des erreurs dans son code, le programme renvoie une erreur claire de manière à ce que l'utilisateur puisse facilement identifier leur erreur.

La dernière correction a été de documenter notre test et de le rendre plus compréhensible en ajoutant des commentaires pour expliquer ce que fait chaque fonction de test et en ajoutant également les noms des auteurs comme demandé.

Question 2 : SIBLINGS

Tout d'abord, nous avons un fichier qui contient les noms, prénoms et âges des personnes écrits sous le format suivant:

```
Wandja!Yvan!20
Hirwa!Raissa!18
Ingenzi!Vany!19
```

Avant les reviews de nos collègues, la fonction :

- devait imprimer un tableau des tous les personnes dans le fichier sous forme de 3 colonnes, sans “!” et “\n” avec comme titre; Noms, Prénoms et âges:

Surname	First Name	Age
Hirwa	Raissa	18
Ingenzi	Vany	19
Wandja	Ivan	18

- En plus de celle-là la fonction devait retourner une liste des prénoms triés par ordre croissant en fonction de leur âges.

["Raissa", "Vany", "Ivan"]

Pour ce faire, l'utilisateur devait aller se documenter sur l'utilisation de la méthode *format* sur les *strings* en python qui comment qui devraient imprimer sous forme de 3 parfaites colonnes même dans le cas où les noms ou prénoms étaient trop longs ou trop courts, ce qui était la première chose négative. Donc les reviews du prof disent de plus simplement demander à l'élève à retourner quelque chose au lieu d'imprimer et retourner.

Deuxième chose négative était que l'énoncé était en anglais mais les spécifications en français ce qui portait confusion pour certains. Du coup, nous avons tous mis en français.

Troisième chose négative était les fautes d'orthographe dans les énoncés. Nous avons essayé de corriger le maximum les fautes d'orthographe. Nous sommes, tous les 3, à la base issus d'une éducation anglophones.

Après les reviews, l'énoncé demande d'ouvrir le fichier, de lire le contenu et de retourner une liste qui contient le prénom et l'âge, sous forme de tuple, de chaque personne dans le fichier.

Si le fichier n'existe pas, la fonction doit lever un `FileNotFoundError`

Si la liste est vide, et que c'était juste une blague, la fonction doit lever un `ValueError`

Si le contenu du fichier est écrit dans le mauvais format, la fonction doit retourner " Le fichier contient des erreurs et donc ne peut pas être lu."

Question 3 : GROCERY SHOPPING:

Voici les divers changement effectué sur cet exercice depuis la lecture des reviews :

- **Changement de la langue de l'énoncé.** L'énoncé initiale était écrite en anglais, donc nous avons du changer la langue utilisé pour faciliter la compréhension. Toujours dans ce contexte nous avons ajouté des exemples pour qu'élève puisse directement voir à quoi nous attendons de son implémentation.
- **Diminuer la difficulté.** Pour certaines fonctions la spécifications ne donnent pas directement une idée de comment les implémenter, donc nous avons opté pour des méthodes dont l'implémentation est plus intuitif sans pour autant changer le contenu de l'exercice.
- **Changer les specifications des méthodes.** Comme cité dans le point précédente, les spécifications n'était pas clair donc nous avons choisi de le rendre le plus clair possible sans évidemment dévoiler la réponse.
- **Les feedbacks.** Pour certaines tests le feedback n'était pas adaptés, ainsi nous avons essayé de rendre le plus spécifique possible les feedbacks.
- **Les cas critiques.** Vu que l'exercice est Objet Oriented, il y a quelques cas critiques soulevés par certaines reviews nous les avons ajouté dans la spécification et nous avons trouvé une manière de les traiter.

REVIEWS

=====

LINFO1002-2020 Review #15A

Paper #15: LINFO1002 P1 A7
Reviewer: Olivier Bonaventure
<olivier.bonaventure@uclouvain.be>

Les énoncés des exercices: 2. Contiennent quelques
erreurs/imprécisions

Les tests unitaires: 3. Fonctionnent pour tous les
exercices mais donnent un feedback
partiel

Le code python de la suite de tests: 3. Certains parties sont difficiles à
comprendre

La diversité des exercices: 3. Les trois exercices portent sur des
parties différentes de la matière

Appréciation globale: 3. Le travail global est bon, il
mérite une cote entre 12 et 16

===== Commentaires pour les auteurs =====

Replace

Pensez à fournir un exemple clair d'utilisation de votre fonction dans l'énoncé, cela rendra la question plus facile à comprendre.

Fournissez des spécifications avec pre et post comme dans le premier cours. Indiquez comment la fonction doit réagir si les spécifications ne sont pas respectées.

Pensez à vérifier l'orthographe de vos questions.

N'oubliez pas de documenter correctement votre suite de test. C'est un programme python à documenter comme les autres programmes python. Indiquez le nom de l'auteur de la suite de test dans le code python.

Votre test python est fort simple. Vous devriez imaginer les erreurs que les étudiants pourraient faire et construire des tests spécifiques qui permettent de détecter ces erreurs et ensuite suggérer un message de feedback utile.

Cette question est assez classique

Siblings

Question intéressante sur le parsing des fichiers. L'énoncé est assez clair. Pour l'ordre croissant, indiquer que c'est l'ordre croissant des âges.

Faites une fonction qui retourne un string plutôt qu'une fonction qui affiche, ce sera beaucoup plus facile à tester, mais c'est une bonne idée d'essayer de trouver des solutions sur stackoverflow

Que doit faire l'étudiant si une ligne du fichier ne respecte pas le format ?

Votre code python fournit un feedback détaillé. Pensez à effacer les fichiers à la fin des tests

Grocery Shopping

Soyez un peu plus précis sur les types de variables d'instances des classes

Fournissez un exemple d'utilisation de vos classes. Il y a beaucoup à écrire pour l'étudiant, mais c'est une illustration intéressante de l'utilisation d'objets.

Pour Product, que se passe-t-il lorsque l'on fait buy pour un produit dont la quantité est 0 ?

Les objets Ray et Shop pourraient être un peu mieux expliqués

N'oubliez pas de documenter correctement votre suite de test. C'est un programme python à documenter comme les autres programmes python. Indiquez le nom de l'auteur de la suite de test dans le code python.

=====

LINFO1002-2020 Review #15B
Updated 4 Mar 2020 09:07:09 CET

Paper #15: LINFO1002 P1 A7

Les énoncés des exercices: 1. Contiennent de trop nombreuses
erreurs/imprécisions
Les tests unitaires: 2. Fonctionnent pour certains
exercices mais pas tous
Le code python de la suite de tests: 4. Est clair et compréhensible
La diversité des exercices: 3. Les trois exercices portent sur des
parties différentes de la matière
Appréciation globale: 1. Le travail global est insuffisant,
il mérite une cote inférieure à
10

===== Commentaires pour les auteurs =====

L'énoncé du premier exercice contient quelques typos et sa langue n'est pas cohérente. La description parle de caractères ou mots mais la signature ne mentionne que des mots. La réponse est en principe très simple mais un code comme "return ".join(c if c != char_1 else char_2 for c in text)" ne passe pas les tests.

L'énoncé du deuxième exercice contient également quelques typos. Vous demandez à l'étudiant de retourner une erreur, hors en Python les erreurs sont soulevées avec le mot clé "raise". Vous demandez également de "Si la liste n'est pas vide, la fonction doit imprimer une liste des éléments du fichier" hors dans l'exemple que vous donnez, ou vous imprimez le retour de la fonction, on voit ces éléments s'afficher.

Les tests affichent "Votre fonction retourne <module 'StudentCode' from '/task/student/StudentCode.py'> != <module 'CorrCode' from '/task/student/CorrCode.pyc'>, ce dernier étant le résultat attendu", quel est l'intérêt de ce test ?

Relisez également l'énoncé du dernier exercice. Le mot "shelf" est le bon mot traduisant "rayon" dans votre contexte. Vous avez spécifié les méthodes, c'est un bon début mais certains cas ne sont pas pris en compte. Que se passe-t-il si l'on achète un produit dont la quantité est 0 ?

"si product est un objet de Product, et de faire telle sorte que le lst_products soit ordonner d'ordre croissant." et si ce n'est pas le cas, où doit se trouver cet objet dans la liste?

Vos tests semblent conséquent, je n'ai pas eu le temps de les essayer. Ils contiennent quelques typos par contre.

Vos deux premiers exercices doivent absolument être améliorés, et peut-être aussi complexifiés.

=====

LINFO1002-2020 Review #15C

Paper #15: LINFO1002 P1 A7

Les énoncés des exercices: 3. Sont clairs et non-ambigus

Les tests unitaires: 3. Fonctionnent pour tous les
exercices mais donnent un feedback
partiel

Le code python de la suite de tests: 3. Certains parties sont difficiles à
comprendre

La diversité des exercices: 3. Les trois exercices portent sur des
parties différentes de la matière

Appréciation globale: 3. Le travail global est bon, il
mérite une cote entre 12 et 16

===== Commentaires pour les auteurs =====

ex1: l'énoncé est clair

lorsque qu'il y a une erreur, les tests ne montre pas le char_1 et le char_2 --> difficile de
voir où se trouve la faute

les tests sont complets

faute de frappe "Retourner une text modifie" (dans les post conditions)

ex2: durant l'examen, on n'a pas accès à internet donc le lien vers la doc n'est pas un bon
entraînement pour l'examen

l'utilisation des fichiers txt me pose plusieurs questions :

1) est-ce que vous avez prévu le cas où l'étudiant (au lieu de le lire) écrit quelque
chose dans le fichier? (j'ai pas testé pour éviter de tout casser)

2) si plusieurs étudiants font le test en même temps et qu'ils ne font le
"file.close()", est-ce qu'il y a pas de bug causés?

les tests sont clairs et complets

l'énoncé est clair, ce serait encore plus clair si les exemples étaient à chaque fois avec
les mêmes données ("Bean!Mister!54" et "Tunga Olive-Mihigo 3")

ex3: il y a moyen de mettre des bouts de code dans la partie pour que l'étudiant puisse le
modifier directement

il manque de nombreux ":" après les en-têtes des fonctions ("def():")

la méthode buy() n'a pas de pré-ordres dans les tests, il faut que la quantité soit ≥ 1 pour
appliquer la post condition

le test de la fonction sorting_product est pas correct pcq vérifie l'objet et non son
contenu. (il manque le def__eq__():)

faute de frappe dans le test avec le GSM : "n'oublie pas de trier les rayons"

=====

Les énoncés des exercices: 2. Contiennent quelques erreurs/imprécisions

Les tests unitaires: 3. Fonctionnent pour tous les exercices mais donnent un feedback partiel

Le code python de la suite de tests: 4. Est clair et compréhensible

La diversité des exercices: 3. Les trois exercices portent sur des parties différentes de la matière

Appréciation globale: 2. Le travail global est satisfaisant, il mérite une cote entre 10 et 12

===== Commentaires pour les auteurs =====

C'est étonnant que les énoncés des exercices soient entièrement en anglais alors que la spécification des fonctions est en français.

Exercice Replace:

L'exercice est trop simple. Il suffit de savoir utiliser la méthode `str.replace` de python. Ne pas autoriser l'étudiant à utiliser la méthode ``replace`` serait une manière de rendre l'exercice plus intéressant.

Exercice Siblings:

Les postconditions telles qu'affichées ne collent pas à celles du correctif ou même à celles de l'énoncé ; Selon les postconditions écrites sur INGINIOUS, la fonction doit retourner soit `ValueError` soit "la liste des ages en ordre croissant" or le contexte de la question dit "return a list of theirs[sic] names". Quand vous dites "names", est-ce que vous voulez parler des "First Name" ou des "Surname"? Il y a là une ambiguïté. La fonction du correctif retourne une liste des "Surnames", ce qui voudrait dire qu'il faut plutôt croire au contexte de la fonction et déterminer que "names" fait référence aux surnames, mais ne serait-ce pas une erreur? Dans les dernières lignes du code du correctif, une liste ``age`` est initialisée pour soit-disant stocker l'age de chaque personne si on en croit aux commentaires mais en réalité, la liste ``age`` n'est pas une liste des âges, mais une liste des surnames, ce qui est tout à fait contre-intuitif et relève sûrement d'une erreur.

Par ailleurs, que voulez-vous dire par "ordre croissant"? Selon l'ordre alphabétique des "First Name"? Selon l'ordre anti-alphabétiques des "Surnames"? Selon l'ordre croissant des entiers des "Age"? Cela ne ferait pas de mal de répéter que c'est selon l'âge comme il a été fait dans le contexte de la question.

Notes sur la fonction correcte (Projet_2/q2_Siblings/src/CorrCode.py):

À la ligne 19, vous mettez un ``return`` à la suite d'un ``raise`` dans l'optique, je suppose, de mettre la fin à l'exécution de la fonction mais ce n'est pas nécessaire puisque le ``raise`` s'en charge. La ligne 19 est en fait jamais atteinte par le programme.

À la ligne 37, l'"[ajout] de l'age de chacun dans la liste age" a été mal faite car c'est en fait le surnom qui est ajouté à la liste ``age``. Il faudrait changer cette ligne en ``age.append(tab[i][2])``

La boucle qui commence à la ligne 27 semble utiliser un tri à bulles qui est loin d'être idéal comme algorithme de tri (cf. cours d'introduction à l'algorithmique).

Exercice Grocery Shopping:

La traduction anglaise de "rayon de magasin" ce n'est pas "ray". "Ray" c'est dans le sens rayon de lumière. Le mot que vous cherchez est "aisle".

Le `test4` (`test4_ray_sorting`) compare 2 objets `Product` mais ces objets n'ont pas de représentation en forme de string. Du coup, dans le cas où une différence existe entre l'objet de l'étudiant et celui du correctif, le feedback affiche la référence de l'objet dans la mémoire (`<Studentcode.Product object at 0x49574398543..`). Ce n'est pas vraiment grave mais ce serait sympa si l'objet `Product` pouvait avoir une représentation plus lisible.

J'ai personnellement eu du mal à comprendre les spécifications des méthodes.

=====

LINFO1002-2020 Review #15E

Paper #15: LINFO1002 P1 A7

Les énoncés des exercices: 2. Contiennent quelques
erreurs/imprécisions

Les tests unitaires: 4. Fonctionnent pour tous les
exercices et fournissent un
feedback clair

Le code python de la suite de tests: 3. Certaines parties sont difficiles à
comprendre

La diversité des exercices: 3. Les trois exercices portent sur des
parties différentes de la matière

Appréciation globale: 3. Le travail global est bon, il
mérite une cote entre 12 et 16

===== Commentaires pour les auteurs =====

Replace:

L'exercice est peut-être un peu simple, je ne vois pas d'autres solutions que d'utiliser la fonction `replace`, ça aurait pu être intéressant d'y ajouter des nuances pour forcer l'élève à réfléchir plus loin que les fonctions pré-implémentées.

Sibling:

L'idée de demander à l'élève de créer un tableau plutôt qu'une liste ou un dictionnaire ajoute de la difficulté à l'exercice et c'est intéressant.

C'est bien d'avoir mis des commentaires dans les codes, mais pour le code test ça aurait pu être pratique de le faire dans tout le code plutôt que juste dans la première petite partie.

Grocery:

L'exercice aurait pu être un peu mieux expliqué.

=====

LINFO1002-2020 Review #15F

Paper #15: LINFO1002 P1 A7

Les énoncés des exercices: 2. Contiennent quelques
erreurs/imprécisions

Les tests unitaires: 3. Fonctionnent pour tous les
exercices mais donnent un feedback
partiel

Le code python de la suite de tests: 3. Certaines parties sont difficiles à
comprendre

La diversité des exercices: 3. Les trois exercices portent sur des
parties différentes de la matière

Appréciation globale: 3. Le travail global est bon, il
mérite une cote entre 12 et 16

===== Commentaires pour les auteurs =====

Pour la question "Replace":

Au niveau de l'utilisateur, l'énoncé n'est pas tout à fait clair pour moi. Un exemple aurait pu m'aider à trouver la bonne solution, car avec la spécification donnée j'ai une erreur de sémantique. Attention aussi à la grammaire et ce serait mieux de tout écrire dans une langue. Les tests ont un message identique: "Votre fonction `replace` n'est pas implémentée de la meilleure manière" qui ne donnent pas une indication un peu plus ciblée sur ce qui ne va pas avec mon code.

#####

txt = ""

for i in text:

```
if i == char_1:
    txt += char_2
txt += i
return txt
#####
[Submission #5e65e1986779dd5919d21a36]
```

Même en supprimant un bout de mon code pour provoquer une autre erreur, j'obtiens quasi le même feedback sans beaucoup d'explication. Je pense que les pre ne sont pas assez claires. Je ne sais pas si par "caractère" on désigne un caractère du texte, ou une substring. C'est là qu'un exemple donné aurait été utile.

Au niveau de la suite de test, le code est assez court (seulement 2 tests) et les noms de variables ne sont pas représentatifs sachant qu'il n'y a pas de commentaires. Le message du feedback n'aide pas beaucoup. Bonne utilisation du bloc "try..except" qui capture les erreurs en python.

Pour la question "Siblings":

Au niveau de l'utilisateur, l'énoncé est assez clair et les exemples donnés aident à comprendre la tâche à réaliser. Cependant, il a quelques erreurs sur la forme et vous vous êtes trompés sur certains mots dans les post (liste au lieu de file).

Votre fonction doit retourner une liste, mais vous conseillez de faire des print, donc il y a un conflit dans les consignes à ce niveau là.

Au niveau de la suite de test, le code est bien structuré et les exceptions sont bien gérées. Les commentaires aident à s'y retrouver et le message de feedback pour chaque test est assez clair et utile pour l'élève.