

ITI 1520 – Test 1

Disponible: le 3 octobre, 2020, 12:00

Durée: 1 heure et 30 minutes, plus 15 min pour le remettre.

SVP notez que le test n'est pas accepté après cette date, le système BrightSpace va fermer à 13:45.

Notes:

Vous devez faire ce travail **individuellement**. Vous devez soumettre seulement le fichier test1.py en BrightSpace. Il y a 4 questions (4 fonctions requises). Il faut mettre toutes les fonctions dans un fichier test1.py

Ajoutez des commentaires pour chaque fonction (seulement le contrat de type, déterminé par vous-même à base de la description des fonctions).

Tester avec programme principale ou dans l'interpréteur Python pour vous-même, pas besoin de soumettre des tests.

Pendant le test, c'est permis d'utiliser les notes disponibles en BrightSpace (notes de cours, de labo, devoirs), le manuel de cours, et documentation de python.org et de IDLE.

Si vous utilisez autre chose, il faut ajouter une déclaration dans votre soumission en BrightSpace et donner la référence dans un fichier declaration_VOTRE_NOM.txt. Voir ci-dessous pour plus des détails.

Le nom de fichier et les noms des fonctions doivent être les noms requis, parce qu'on va utiliser des tests automatiques pour la correction. Si le nom d'une fonction n'est pas le nom requis, la note sera zéro pour la fonction.

Si le fichier .py donne erreur de syntaxe, la note sera zéro pour les questions dans le fichier.

Si vous envoyez le test plusieurs fois, la dernière version sera notée.

SVP noter qu'on va utiliser un outil logiciel pour détecter du plagiat. En cas de deux tests identiques ou très similaires, la note sera zéro pour les deux.

Pendant le test ITI1520 les serveurs discord sur Internet seront surveillés par les assistants.

Barème : total de 40 points. Test 1 est 11% de la note finale.

Question 1. (10 points) `atlantic(sn)`

Créez une fonction en Python nommée `atlantic` qui prend comme entrée une chaîne de caractères (appelée `sn`, par exemple), et retourne une autre chaîne de caractères. Si la chaîne de caractères donnée comme entrée contient 9 caractères ou plus, retournez la chaîne "nouveau", sinon retournez la chaîne "vieux". Pour calculer le nombre de caractères utilisez `len(sn)`.

Exemples des tests dans l'interpréteur :

```
>>> atlantic("12345678988")
'nouveau'
>>> atlantic("300056789")
'nouveau'
>>> atlantic("8899234")
'vieux'
>>> atlantic("today")
'vieux'
```

Question 2. (10 points) `southern(n)`

Créez une fonction en Python nommée `southern` qui prend comme entrée un entier `n` de valeur 1 ou 2 (pas besoin de vérifier si la valeur de `n` n'est pas 1 ou 2, on fait l'hypothèse que la valeur donnée est correcte). Si `n` est 1, la fonction demande l'utilisateur d'entrer le poids en livres et onces (du clavier) et affiche un message avec le poids transformé en kilogrammes. Utilisez la formule de conversion: $\text{kilogrammes} = (\text{livres} * 16 + \text{onces}) * 0.02835$

Si `n` est 2, la fonction demande l'utilisateur d'entrer son nom et son numéro d'étudiant. Si le numéro d'étudiant à 9 chiffres ou plus, la fonction affiche un message qui dit que le numéro est nouveau, sinon, qu'il est vieux. Pour voir s'il faut afficher vieux ou nouveau, il faut appeler la fonction `atlantic` de Question 1.

Exemples des tests dans l'interpréteur :

```
>>> southern(1)
Entrez le poids en livres: 1
Entrez le poids en onces: 2.2
1.0 livres et 2.2 onces est (approx.) 0.5159699999999999 kilogrammes.
>>> southern(1)
Entrez le poids en livres: 55.235
Entrez le poids en onces: 3.5
55.235 livres et 3.5 onces est (approx.) 25.153821 kilogrammes.
>>> southern(2)
Quel est ton nom? Champion
Quel est ton numero d'etudiant? 300056789
Champion ton numero d'etudiant est nouveau
>>> southern(2)
Quel est ton nom? Marie
Quel est ton numero d'etudiant? 8878654
Marie ton numero d'etudiant est vieux
```

Question 3. (10 points) `pacific(g1, g2, g3)`

Considérez un système de notes de 0 à 100 (0 et 100 sont incluses). Une note supérieure ou égale à 50 est une note de passage. Une note supérieure ou égale à 80 est un A.

Créez une fonction en Python nommée `pacific` qui prend comme entrée trois nombres réels de 0 à 100 (inclusive). Vous n'avez pas besoin de tester si les nombres sont dans cet intervalle (on fait l'hypothèse que c'est le cas). Les trois nombres `g1`, `g2`, `g3` sont les notes d'un étudiant pour 3 cours. La fonction doit retourner `True` si toutes les trois notes sont des notes de passage et au moins une est un A. Sinon, la fonction doit retourner `False`. Pour cette question il ne faut pas utiliser des `if`. Utilisez des expressions booléennes. Si vous utilisez des `if`, vous allez perdre 20% des points pour cette question.

Exemples des tests dans l'interpréteur :

```
>>> pacific(49.8, 82, 70.5)
False
>>> pacific(55, 82, 70.5)
True
>>> pacific(80.1, 82, 70.5)
True
```

Question 4. (10 points) `arctic(n)`

Un nombre est un palindrome si la séquence des chiffres de gauche à droite est la même que la séquence de droite à gauche. Outrement dit, le nombre est un palindrome si la première chiffre est égale avec la dernière, la deuxième chiffre est égale avec l'avant dernière, la troisième chiffre est égale avec la troisième de la fin, etc. Par exemple, 123321 est un palindrome, mais 123421 n'est pas un palindrome.

Créez une fonction en Python nommée `arctic` qui prend comme entrée un entier positif `n` qui a 4 ou 6 chiffres. (On fait l'hypothèse que la fonction sera utilisée seulement avec des entiers positifs de 4 ou 6 chiffres.) La fonction doit retourner `True` si `n` est un palindrome et `False` sinon.

Pour cette question il ne faut pas utiliser des chaînes des caractères. Il faut utiliser modulo et division entière. Si vous utiliser des chaînes des caractères pour tester si `n` a 4 ou 6 chiffres, vous allez perdre 20% des points pour cette question. Si vous utilisez des chaînes des caractères aussi pour vérifier si c'est un palindrome, vous allez perdre encore 20%.

Exemples des tests dans l'interpréteur :

```
>>> arctic(5225)
True
>>> arctic(123123)
False
>>> arctic(3123)
False
>>> arctic(434434)
True
```

Notes pour le fichier `declaration_VOTRE_NOM.txt` :

Ce fichier doit contenir la référence pour code qui n'est pas écrit par vous même **si c'est le cas**. Ça inclue code donné par un collègue ou autre personne, ou trouvé sur l'internet, réseaux sociale (comme Stack Overflow, discord, ou autre). Ça n'inclue pas le code donné en BrightSpace dans les notes de classe, labo, etc.

Pour chaque question ou vous avez utilisé du code donné ou trouvé, il faut:

1. le numéro de question
2. copier-coller le code emprunter. Ça inclue code donné/trouvé et modifier très peu.
3. le nom de la source: personne, site Internet ou autre

Vous allez perdre des points la question, mais au moins vous évitez une accusation de plagiat. En cas de plagiat, la note est zéro et le cas doit être rapporté au doyen. Le même est valable si quelqu'un montre son code a un collègue.

Si vous n'utilisez pas cette déclaration, c'est équivalent à déclarer que tous le code a été écrit par vous-même.

Version en anglais de questions, pour les étudiants qui ont besoin :**Q1. `atlantic(sn)` - 10 points**

Write, a Python function called `atlantic` that takes as input a string, `sn`. If `sn` has 9 or more characters, the function should return a string 'new', otherwise it should return a string 'old'.

Q2. `southern(n)` - 10 points

Write, a Python function called `southern` that takes as input integer, `n` where `n` is 1 or 2. (You may assume that `n` will be 1 or 2, i.e. your function will not be tested with values for `n` different from 1 and 2). If `n` is 1 the function should prompt the user for the weight in pounds first and then weight in ounces, and then it should print a message about the total weight in kilograms. Here is the conversion formula:

$\text{kilograms} = (\text{pounds} * 16 + \text{ounces}) * 0.02835$

If `n` is 2, the function should prompt the user for the name and then for his/her student number. If the student number has at least 9 digits, a message should be print informing the student that they have new uOttawa student number, otherwise that they have old uOttawa student number. In this part (`n` is 2 part) you must make a call to `atlantic` function from Question 1.

Q3. `pacific(g1, g2, g3)` - 10 points

Consider a grading system where for each course students get a numerical grade between 0 to 100 (inclusively). A numerical greater or equal to 50 is considered passing grade. In addition, a numerical grade greater or equal to 80 is also an A.

Write, a Python function called `pacific` that takes as input three numbers each between 0 to 100 (inclusively). (You may assume that your function will only be tested with numbers between 0 and 100 inclusively.) The three numbers, `g1`, `g2`, `g3` represent numerical grades in 3 courses. The function should return `True` if all three grades are passing and at least one is A. Otherwise the function should return `False`. In this question you are not supposed to use `if` statements. If you use `if` statements, you will loose at least 20% of the grade for this question.

Q4. `arctic(n)` - 10 points

A number is called palindromic if the sequence of digits from left to right is the same as from the right to left. Equivalently, a number is palindromic if the first digit is equal to the last, and, if the second digit is equal to the second from the back, and if the third digit is equal to the third from the back ... and so on. For example 123321 is palindromic, but 123421 is not.

Write, a Python function called `arctic` that takes as input an integer, `n`, where `n` can only be a 4-digit or 6-digit positive integer. (You may assume that your function will only be tested with 4-digit or 6-digit positive integers.) The function should return `True` if `n` is palindromic and otherwise it should return `False`. In this question you are not supposed to use strings. If you use strings only to figure out if the number is 4 or 6 digit, you will loose 20% of your grade. If you use it to check palindromic requirement, you will loose additional 20% of the grade for this question.