[redacted]

Commencé le	vendredi 7 octobre 2022, 09:32
État	Terminé
Terminé le	vendredi 7 octobre 2022, 11:30
Temps mis	1 heure 57 min
Points	15,63/20,00
Note	**19,53** sur 25,00 (**78,13**%)

Question **1**

Terminé

Non noté

Sur mon honneur, j’ affirme compléter cet examen par moi-même, sans communication avec personne, et selon les directives identifiées sur la première page de l’examen.

Afin de ne pas créer de situations inégales pour les futures cohortes, j' affirme ne pas copier et partager le contenu de cet examen.

Une fois ma note obtenue et mes réclamations faites et traitées, j' affirme supprimer tout le matériel que j'ai créé ou copié afin de répondre aux questions dans les 30 jours qui suivent l'obtention de ma note.

Votre réponse est correcte.

La réponse correcte est :
Sur mon honneur, j’[affirme] compléter cet examen par moi-même, sans communication avec personne, et selon les directives identifiées sur la première page de l’examen.

Afin de ne pas créer de situations inégales pour les futures cohortes, j'[affirme] ne pas copier et partager le contenu de cet examen.

Une fois ma note obtenue et mes réclamations faites et traitées, j'[affirme] supprimer tout le matériel que j'ai créé ou copié afin de répondre aux questions dans les 30 jours qui suivent l'obtention de ma note.

Question **2**

Partiellement correct

Note de 1,00 sur 1,50

Combien de tests sont-ils nécessaires pour avoir les couverture des instructions (Statement coverage), branches, et conditions des de ce bloc:

```
If(a && b) {
    doSomething();
} else if (d || e) {
    DoSomethingElse();
```

```
} else {
    doAnotherThing();
}
```

Couverture d'instructions (combien de test pour couvrir l'ensemble des instructions (Statement coverage)):
3 ✔

Couverture de branches (combien de test pour couvrir l'ensemble des branches): 3 ✔

Couverture de conditions (combien de test pour couvrir l'ensemble des conditions): 4 ✖

Question **3**
Correct
Note de 0,50 sur 0,50

Pour tout programme, il est généralement possible de trouvé tous les tests qui seraient nécessaires pour s'assurer que notre programme n'a pas de défauts.

Veuillez choisir une réponse.
- Vrai
- Faux ✔

La réponse correcte est « Faux ».

Question **4**
Correct
Note de 0,50 sur 0,50

Il faut attendre qu'un code de programme soit complet pour commencer les activités de test.

Veuillez choisir une réponse.
- Vrai
- Faux ✔

Nous pouvons tester les spécifications avant même qu'une seule ligne de code ne soit testée.

La réponse correcte est « Faux ».

Question **5**
Correct
Note de 0,50 sur 0,50

Selon Weak Equivalence Class Testing (WECT), compte tenu d'un programme avec cinq valeurs (V, W, X, Y, Z) et les classes d'équivalence suivantes, combien de cas de test seraient nécessaires ?

Classes d'équivalence de la valeur V:

V1 = classe_eq_1_de_V
V2 = classe_eq_2_de_V

Classes d'équivalence de la valeur W:

W1 = classe_eq_1_de_W
W2 = classe_eq_2_de_W

Classes d'équivalence de la valeur X:

X1 = classe_eq_1_de_X
X2 = classe_eq_2_de_X
X3 = classe_eq_3_de_X

Classes d'équivalence de la valeur Y:

Y1 = classe_eq_1_de_Y
Y2 = classe_eq_2_de_Y

Classes d'équivalence de la valeur Z:

Z1 = classe_eq_1_de_Z
Z2 = classe_eq_2_de_Z

Réponse : 3 ✔

Nous pouvons modifier tous les entrées des classes d'équivalence à la fois dans WECT.
ex:
cas1= W1,X1,Y1,Z1
cas2= W2,X2,Y2,Z2
cas3= W2,X3,Y2,Z2

3 cas couvrent donc tous les entrées.

La réponse correcte est : 3

Question **6**
Correct
Note de 0,50 sur 0,50

Les activités de test ne sont pas nécessaires pour des systèmes de petite taille (moins de 1000 LOC) développés en Java.

Veuillez choisir une réponse.

- ○ Vrai
- ◉ Faux ✔

La réponse correcte est « Faux ».

Question **7**
Correct
Note de 0,50 sur 0,50

Associez les termes à leurs définitions/exemples.

Défaillance	Cessation de l'aptitude d'un produit à accomplir une fonction requise. ✔
Défaut	Bogues dans un logiciel, incohérence dans les caractéristiques d'un logiciel, mauvais choix dans la conce ✔
Erreur	Action humaine qui produit un résultat incorrect ✔

Votre réponse est correcte.

La réponse correcte est :
Défaillance → Cessation de l'aptitude d'un produit à accomplir une fonction requise.,

Défaut → Bogues dans un logiciel, incohérence dans les caractéristiques d'un logiciel, mauvais choix dans la conception...,

Erreur → Action humaine qui produit un résultat incorrect

Question **8**

Incorrect

Note de 0,00 sur 0,50

Lequel des suivants est un mutant valide et non-equivalent pour la ligne 5:

```
1 def print_somme(a, b):
2     result = a + b
3     if result > 0 and a != 0:
4         print("vert ", result)
5     elif (a + b) < 0 or b != 0:
6         print("red", result)
7 print("fini")
```

- a. elif (a + b) > 0 or b >0: ✖
- b. elif (a + b) < 0 or b !=0:
- c. elif (a + b) < 0 or c !=0:
- d. elif (a + b) < 0 or b <0:
- e. elif (result) < 0 or b !=0:

Votre réponse est incorrecte.

La réponse correcte est : elif (a + b) < 0 or b <0:

Question **9**

Correct

Note de 0,50 sur 0,50

Étant donné la fonction ci-dessous, combien de cas de test seraient nécessaires pour tester la fonction de manière exhaustive?

```
float add(float var1, float var2){
    return var1 + var2;
}
```

Supposez que:
les entiers ont 32 bits
les flottants ont 32 bits
les doubles ont 64 bits

*Malheureusement, les signes de multiplication et les exposants ne sont pas autorisés dans la boîte de réponse. Donc, si votre réponse est très grande, vous devez exprimer votre réponse en utilisant la notation e. (par exemple, 5*10^2 serait 5e2)

Réponse : 2e64 ✖

La réponse correcte est : 1,8446744E+19

Commentaire :

Question **10**

Question 10

Partiellement correct

Note de 0,38 sur 0,50

*Question basée sur la Question 110 créée dans le cours 05.
Sélectionnez le type correct (DEF, C-USE, P-USE) pour la variable sélectionnée dans les instructions suivantes.

La variable answer est un: C-USE ✔ dans l'instruction: return answer

La variable y est un: P-USE ✔ dans l'instruction: if (table[y+1] >5)

La variable z est un: DEF ✖ dans l'instruction: z == 7

La variable upper est un: DEF ✔ dans l'instruction: upper = a.length

Description

En utilisant le programme suivant

```
1    public static void search(String txt, String pat)
2    {
3        int M = pat.length();
4        int N = txt.length();
5        for (int i = 0; i <= N - M; i++) {
6            int j;
7            for (j = 0; j < M; j++)
8                if (txt.charAt(i + j) != pat.charAt(j))
9                    break;
10           if (j == M)
11               System.out.println("Pattern found at index " + i);
12       }
13   }
```

Question 11

Correct

Note de 1,50 sur 1,50

En utilisant le code ci-dessus, complétez les DC-PATH suivants:

DC-PATH(j, 7 ✔ , 10)

DC-PATH(i ✔ , 5, 8)

DC-PATH(N, 4, 5 ✔)

Question 12

Correct

Note de 0,50 sur 0,50

Pour le problème ci-dessus, quelle réponse ne contient **pas** un DU-Path pouvant être utilisé pour satisfaire le critère all-DEF?

a. DU-PATH(M, 3, 5)

b. DU-PATH(txt, 4, 8) ✔ Ne commence pas par un nœud DEF pour "pat"

c. DU-PATH(j, 7, 8)

d. DU-PATH(txt,1,4)

Votre réponse est correcte.

La réponse correcte est :
DU-PATH(txt, 4, 8)

DU PATH(txt, 4, 8)

Question **13**

Terminé

Non noté

Vous pouvez utiliser cette case pour écrire toutes les hypothèses (le cas échéant) que vous avez utilisées pour répondre à cette question.

pour le j à la ligne 7 mon hypothèse est que pour la dernière itération quand j=M le DC-PATH va être valide, car le j++ ne va pas s'exécuter donc il n'y aura pas de redéfinition et on aura un USE à la ligne 10.

Question **14**

Correct

Note de 0,50 sur 0,50

L'analyse des valeurs limites est une technique de test de la boîte [noire] ✔ .

L'analyse des valeurs limites fonctionne bien quand les variables ont une plage de valeurs [limitées] ✔ .

Elle est donc, [mal adaptée] ✔ pour tester un programme qui prend en entrée des chaînes de caractères.

[bien adaptée]

[blanche] [grise]

[aléatoires] [illimitées]

Votre réponse est correcte.

Quand on dit une chaîne de characteres, nous ne connaissons pas nécessairement la limite de la chaîne. Généralement, c'est donc une entrée sans limite fixe, qui est mal adapter pour l'analyse des valeurs limites.

La réponse correcte est :
L'analyse des valeurs limites est une technique de test de la boîte [noire].

L'analyse des valeurs limites fonctionne bien quand les variables ont une plage de valeurs [limitées].

Elle est donc, [mal adaptée] pour tester un programme qui prend en entrée des chaînes de caractères.

Question **15**

Partiellement correct

Note de 0,25 sur 1,50

La fonction sin(x) programmé fonctionne avec en entrée un int entre -180 et 180 degrés et retourne une valeur entre -1.0 et 1.0 ou ERREUR. Complétez les classes d'équivalence suivantes.

Classes d'équivalence de l'entrée

EC1 : x est [*un nombre entier*] ✔

EC2 : x n'est pas [*un nombre flottant*] ✖

Classes d'équivalence du nombre entré valide

EC1.1 : x [=] ✖ -180

EC1.2 : -180 [<] ✖ x [<] ✖ 180

EC1.3: x [=] ✖ 180

un nombre flottant | une valeur

un nombre entier | une valeur

[<] [>]

[=] [<=] [>=] [>]

[>] [<=] [>=] [=]

[<] [>]

Votre réponse est partiellement correcte.

Vous en avez sélectionné correctement 1.

Le nombre de classes d'équivalence est implicite (il n'y a que 2 boites pour EC1 et EC2, et 3 boites pour les classes d'équivalence pour le nombre entré valide (EC1.1, EC1.2 EC1.3).
Les réponses pour EC1 et EC2 doivent compléter les classes d'équivalence de l'entrée. Seulement avec EC1 = "x est un nombre entier" (cas avec succès) et EC2 = "x n'est pas un nombre entier" (cas avec ERREUR) pouvons nous couvrir tous les cas possible. Exemple EC2 : x n'est pas un nombre flottant ne nous donne pas d'information pour ce problème car "pas un nombre flottant" n'est pas une restriction de la fonction.

EC1.1, 1.2, et 1.3 doivent couvrir tous les cas possible pour notre fonction. Il y a trois possibilités: entrée trop petite, entrée correcte, entrée trop grande.

x < -180 nous donne une erreur
-180 <= x <= 180 sont tous des nombres valides (-1 à +1)
x > 180 nous donne une erreur
Selon EC1 tous les int sont des entrées valides et il doivent donc tous êtres considérer.

La réponse correcte est :
La fonction sin(x) programmé fonctionne avec en entrée un int entre -180 et 180 degrés et retourne une valeur entre -1.0 et 1.0 ou ERREUR. Complétez les classes d'équivalence suivantes.

Classes d'équivalence de l'entrée

EC1 : x est [un nombre entier]

EC2 : x n'est pas [un nombre entier]

Classes d'équivalence du nombre entré valide

EC1.1 : x [<] -180

EC1.2 : -180 [<=] x [<=] 180

EC1.3: x [>] 180

Question **16**

Correct

Note de 2,00 sur 2,00

La fonction cercle (p, ray, c) reçoit en entrée

- Un point « p=(x, y) » (int, int)
- Un rayon « ray », (int)
- Une couleur « c » pouvant être ROUGE, JAUNE, BLEU ou BLANC

Selon la couleur spécifiée, la fonction affiche un cercle ayant comme centre « p » et comme rayon « ray ». Si x<0 et y<0 le cercle peut être ROUGE ou JAUNE; si x>0 et y>0 le cercle peut être BLEU ou BLANC, sinon le cercle doit être VERT.

Si les paramètres ne sont pas corrects la fonction retourne « error ».

Complétez les paramètres, choix, et contraintes suivantes, et complétez les trames de tests *pour des valeurs valides de p,ray, et c* avec le **critère Each Choice**:

P :

	Choix	Contraintes
P1	x<0 et y<0	[propriété point négatif]
P2	x>0 et y>0	[propriété point positif]
P3	x<0 et y>0	[propriété point mix]
P4	x>0 et y<0	[propriété point mix]
P5	NON-Réel	[error]
P6	NON-Numérique	[error]

ray :

	Choix	Contraintes
R1	ray>0	[propriété cercle visible] ✔
R2	ray=0	[propriété cercle non visible]
R3	NON-Réel	[error] ✔
R4	NON-Numérique	[error] ✔

C:

	Choix	Contraintes
C1	ROUGE	[si point négatif & cercle visible]

		✔
C2	JAUNE	[si point négatif & cercle visible]
C3	BLEU	[si point positif & cercle visible] ✔
C4	BLANC	[si point positif & cercle visible]
C5	VERT	[si point mix & cercle visible]
C6	NON VALIDE	[error] ✔

Trames de tests à completer pour des valeurs valides pour le **critère Each Choice**:

Trame de test	P	R	C
T-1	P1	R1 ✔	C1
T-2	P1	R1	C2 ✔
T-3	P2	R1 ✔	C3
T-4	P2	R1 ✔	C4 ✔
T-5	P3 ✔	R1 ✔	C5
T-6	P4	R1 ✔	C5

Votre réponse est correcte.

La réponse correcte est :
La fonction cercle (p, ray, c) reçoit en entrée

- Un point « p=(x, y) » (int, int)
- Un rayon « ray », (int)
- Une couleur « c » pouvant être ROUGE, JAUNE, BLEU ou BLANC

Selon la couleur spécifiée, la fonction affiche un cercle ayant comme centre « p » et comme rayon « ray ». Si x<0 et y<0 le cercle peut être ROUGE ou JAUNE; si x>0 et y>0 le cercle peut être BLEU ou BLANC, sinon le cercle doit être VERT.

Si les paramètres ne sont pas corrects la fonction retourne « error ».

Complétez les paramètres, choix, et contraintes suivantes, et complétez les trames de tests *pour des valeurs valides de p,ray, et c* avec le **critère Each Choice**:

P :

	Choix	Contraintes
P1	x<0 et y<0	[propriété point négatif]

P2	x>0 et y>0	[propriété point positif]
P3	x<0 et y>0	[propriété point mix]
P4	x>0 et y<0	[propriété point mix]
P5	NON-Réel	[error]
P6	NON-Numérique	[error]

ray :

	Choix	Contraintes
R1	ray>0	[[propriété cercle visible]]
R2	ray=0	[propriété cercle non visible]
R3	NON-Réel	[[error]]
R4	NON-Numérique	[[error]]

C:

	Choix	Contraintes
C1	ROUGE	[[si point négatif & cercle visible]]
C2	JAUNE	[si point négatif & cercle visible]
C3	BLEU	[[si point positif & cercle visible]]
C4	BLANC	[si point positif & cercle visible]
C5	VERT	[si point mix & cercle visible]
C6	NON VALIDE	[[error]]

Trames de tests à completer pour des valeurs valides pour le **critère Each Choice**:

Trame de test	P	R	C
T-1	P1	[R1]	C1
T-2	P1	R1	[C2]
T-3	P2	[R1]	C3
T-4	P2	[R1]	[C4]
T-5	[P3]	[R1]	C5
T-6	P4	[R1]	C5

Description

*Notez que toutes les étapes de cette question seront évaluées manuellement. Selon le plan de cours:

La note sera basée l'évaluation selon les critères suivants: le choix, la méthodologie, la justification ou explication de la solution retenue, l'exactitude des résultats obtenus, et de la clarté des réponses.

Il est possible d'écrire toutes vos réponses directement sur Moodle, ou de travailler sur votre ordinateur et de uploader un pdf (voir la prochaine section pour la boîte de téléchargement). Si vous avez choisi de uploader un fichier PDF, assurez-vous d'avoir toutes les parties de la question.

Question **17**

Terminé

Non noté

Vous pouvez utiliser cet espace pour écrire tout commentaire/hypothèse (le cas échéant) que vous avez pour la question à développement et toutes ses parties.

Si vous avez choisi de répondre à la question à développement en dehors de Moodle, vous pouvez également utiliser cet espace pour soumettre vos réponses en format pdf à la place.

Je suppose que pour respecter le critère de RWCT, il faut que je test les valeurs à l'extrême des données valides donc temperature<debut_donnees et temperature>fin_donnees, ensuite il faut que je teste les 2 valeurs qui délimitent temperature=5.9 et temperature=8.2. Il faut que je aussi que je teste les valeurs un peu après le minimum et un peu avant le maximum donc temperature=6.1 et temperature=8.0. Pour finir il faut que je teste une valeur 5.9<temperature<8.2.

comme il y a seulement un seul paramètre il faut 7^1 cas de tests.

derrière question un mutant equivalent ne change pas donc test si temp est entre les limites ou égale au limites

Question **18**

Partiellement correct

Note de 3,00 sur 3,50

Le "programme" suivant vous est donné. C'est un programme qui permet à un utilisateur de soumettre une temperature en Celsius (un float (arrondi à une décimale)), et le programme renverra des années où c'était la température moyenne à Montréal. (Liste de String). Votre première tâche consiste à utiliser le critère RWCT (Tests robustes des pire cas) pour développer des cas de test pour la méthode **get_annees_pour_temp**, et de les écrire en test unitaires à l'aide de la bibliothèque unittest de Python.

```
import json

class temp_vs_annee:
    def __init__(self):
        self.debut_donnees = 5.9
        self.fin_donnees = 8.2
        self.data = {}

    def load_data(self):
        with open("year_temp_data.json", 'r') as temp_vs_annee_data_file:
            self.data = json.loads(temp_vs_annee_data_file.read())

    def get_annees_pour_temp(self, temp):
        if temp >= self.debut_donnees:
            try:
                return self.data[str(round(temp,1))]
            except KeyError:
                return "Pas une température valide"

    def calc_temp_spread(self):
        return self.fin_donnees - self.debut_donnees
```

Le fichier year_temp_data.json est accessible ici.

Vous devez avoir maximum un "assert" par test.

*Supposer que toute classe qui utilisera celle-ci effectuera sa propre vérification et essaiera uniquement d'utiliser les types de données appropriés pour les fonctions (par exemple, temp sera toujours un float).

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```
import unittest
#Vous avez access à la classe << temp_vs_annee >> dans code runner, pas néces

class temp_vs_annee_test(unittest.TestCase):
    #...
    def setUp(self):
        self.t_vs_a=temp_vs_annee()

    def test_get_annees_pour_temp_min_debut_donnees(self):
        temp=5.5
        annees=self.t_vs_a.get_annees_pour_temp(temp)
        self.assertIsNone(annees)

    def test_get_annees_pour_temp_debut_donnees(self):
        temp=self.t_vs_a.debut_donnee
        annees=self.t_vs_a.get_annees_pour_temp(temp)
        self.assertEqual(["1965"],annees)

    def test_get_annees_pour_temp_max_debut_donnees(self):
        temp=6.1
        annees=self.t_vs_a.get_annees_pour_temp(temp)
        self.assertEqual(["1964","1969"],annees)
```

	Test	Résultat attendu	Résultat obtenu	
✖	test_OK()	True	False	✖

Votre code doit réussir tous les tests pour gagner des points. Recommencer.

Montrer les différences

Partiellement correct

Note pour cet envoi : 0,00/3,50.

Commentaire :

Q. 18 /3.5 Tests correspondent au criteres RWCT (x/7) 3 Chaque test à un assert adéquat (x/7) 7 Un setup est utilisé(/ 7

Pas de load_data() dans le setup, donc tests de passent pas

Question **19**

Correct

Note de 0,50 sur 0,50

Nous pouvons dire que nous faisons ici des tests en boîte grise ✔ car nous mélangeons boîte noire (RWCT) et boîte blanche (test unitaire avec code source).

Votre réponse est correcte.

La réponse correcte est :
Nous pouvons dire que nous faisons ici des tests en boîte [grise] car nous mélangeons boîte noire (RWCT) et boîte blanche (test unitaire avec code source).

Question **20**

Question **20**

Terminé

Note de 1,25 sur 1,25

En utilisant le programme entier (il est à nouveau présenté ci-dessous) et les tests que vous avez crées. Identifiez un mutant qui **ne survivrait pas** à votre suite de tests.

*Supposer que toute classe qui utilisera celle-ci effectuera sa propre vérification et essaiera uniquement d'utiliser les types de données appropriés pour les fonctions (par exemple, temp sera toujours un float).

```
import json

class temp_vs_annee:
    def __init__(self):
        self.debut_donnees = 5.9
        self.fin_donnees = 8.2
        self.data = {}

    def load_data(self):
        with open("year_temp_data.json", 'r') as temp_vs_annee_data_file:
            self.data = json.loads(temp_vs_annee_data_file.read())

    def get_annees_pour_temp(self, temp):
        if temp >= self.debut_donnees:
            try:
                return self.data[str(round(temp,1)]
            except KeyError:
                return "Pas une température valide"

    def calc_temp_spread(self):
        return self.fin_donnees - self.debut_donnees
```

Dans la case ci-dessous, vous devez :

1) Identifiez le numéro de la ligne de code sur laquelle vous introduisez votre mutant

2) Écrire la ligne de code mutée. Pas besoin de réécrire tout le programme, mais vous devez réécrire la ligne complète, pas seulement l'opérateur de mutation.

1) Numéro de la ligne de code sur laquelle vous introduisez votre mutant:

mutant qui ne survit pas à la ligne 14

2) La ligne de code mutée:

```
if temp < self.debut_donnees:
```

Commentaire :

Question **21**

Terminé

Note de 1,25 sur 1,25

En utilisant le programme entier (il est à nouveau présenté ci-dessous) et les tests que vous avez créés. Identifiez un mutant **qui survivrait** à votre suite de tests.

*Supposer que toute classe qui utilisera celle-ci effectuera sa propre vérification et essaiera uniquement d'utiliser les types de données appropriés pour les fonctions (par exemple, temp sera toujours un float).

```
import json

class temp_vs_annee:
    def __init__(self):
        self.debut_donnees = 5.9
        self.fin_donnees = 8.2
        self.data = {}

    def load_data(self):
```

```
        with open("year_temp_data.json", 'r') as temp_vs_annee_data_file:
            self.data = json.loads(temp_vs_annee_data_file.read())

    def get_annees_pour_temp(self, temp):
        if temp >= self.debut_donnees:
            try:
                return self.data[str(round(temp,1))]
            except KeyError:
                return "Pas une température valide"

    def calc_temp_spread(self):
        return self.fin_donnees - self.debut_donnees
```

Dans la case ci-dessous, vous devez :

1) Identifiez le numéro de la ligne de code sur laquelle vous introduisez votre mutant

2) Écrire la ligne de code mutée. Pas besoin de réécrire tout le programme, mais vous devez réécrire la ligne complète, pas seulement l'opérateur de mutation.

1) Numéro de la ligne de code sur laquelle vous introduisez votre mutant:

mutant a la ligne 14 qui survit à la suite de tests

2) La ligne de code mutée:

```
if temp >= 5.9:
```

Commentaire :
mutant équivalent mais ok oui il survivrait

Question **22**

Partiellement correct

Note de 0,50 sur 2,00

En utilisant votre mutation **qui survivrait à votre suite de tests originale** et le principe des tests de mutation. Créer un nouveau test qui tuerait/éliminerait le mutant. Les tests écrits ici sont exécutés sur le programme sans mutation (ci-dessous) et devrait donc passer.

```
import json

class temp_vs_annee:
    def __init__(self):
        self.debut_donnees = 5.9
        self.fin_donnees = 8.2
        self.data = {}

    def load_data(self):
        with open("year_temp_data.json", 'r') as temp_vs_annee_data_file:
            self.data = json.loads(temp_vs_annee_data_file.read())

    def get_annees_pour_temp(self, temp):
        if temp >= self.debut_donnees:
            try:
                return self.data[str(round(temp,1))]
            except KeyError:
                return "Pas une température valide"

    def calc_temp_spread(self):
        return self.fin_donnees - self.debut_donnees
```

Le fichier year_temp_data.json est accessible ici.

Vous devez avoir maximum un "assert" par test. Vous devez écrire un seul test (test qui éliminerait le mutant)

pour cette partie.

*Supposer que toute classe qui utilisera celle-ci effectuera sa propre vérification et essaiera uniquement d'utiliser les types de données appropriés pour les fonctions (par exemple, temp sera toujours un float).

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```
1  import unittest
2  #Vous avez access à la classe << temp_vs_annee >> dans code runner, pas néces
3
4  class test_temp_vs_annee(unittest.TestCase):
5      #...
6      def setUp(self):
7          self.t_vs_a=temp_vs_annee()
8
9      def test_get_annees_pour_temp(self):
10         temp=4.9
11         self.t_vs_a.get_annees_pour_temp(temp)
12         self.assert
13
```

Erreur(s) de syntaxe

```
  File "__tester__.python3", line 76
    self.assert
               ^
SyntaxError: invalid syntax
```

Partiellement correct

Note pour cet envoi : 0,00/2,00.

Commentaire :

Q. 22 /2.00	Un seul nouveau test	Test avec asser adéquat	Test detecte mutant	Test passe
	0.5	1	0	0	0

◀ Inventaire des questions G02 | Aller à... | Méthodes aléatoires ▶