# INF1007 - Introduction à la programmation



| | |
|---|---|
| Commencé le | jeudi 20 octobre 2022, 10:44 |
| État | Terminé |
| Terminé le | jeudi 20 octobre 2022, 11:04 |
| Temps mis | 19 min 49 s |
| Note | Pas encore évalué |

NAVIGATION DU TEST

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Afficher une page à la fois

Terminer la relecture

Question 1
Terminé
Non noté
Marquer la question

Sur mon honneur, j' affirme je compléter cet examen par moi-même, sans communication avec d'autres personnes.

Afin de ne pas créer de situations inégales pour les futures cohortes, j'

affirme à ne pas copier et partager le contenu de cet examen.

Une fois ma note obtenue et mes réclamations faites et traitées, j'

affirme à 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 d'autres personnes.
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
Correct
Note de 0,50 sur 0,50
Marquer la question

En Python, un nombre décimal (float) est toujours stocké sur 32 bits.

Veuillez choisir une réponse.
Vrai
Faux ✔

La réponse correcte est « Faux ».

Question 3
Correct
Note de 0,50 sur 0,50
Marquer la question

Quelle(s) nom(s) de variable(s) suivante(s) respecte(nt) les règles de nommages des variables?

Sélectionnez la ou les réponse valide.
a. return_2 ✔
b. 5-test
c. -class
d. class
e. _class ✔
f. Class ✔

Votre réponse est correcte.
Les réponses correctes sont :
_class, return_2,
Class

Question 4
Correct
Note de 0,50 sur 0,50
Marquer la question

Dans le code suivant, la ligne 6 ne sera jamais exécutée.

```
1. x = 0
2. while x <= 10:
3.     if x != 10:
4.         break
5.     x += 1
6.     print("EXISTE!!!!")
7.     x += 1
```

Veuillez choisir une réponse.
Vrai ✔
Faux

La réponse correcte est « Vrai ».

Question 5
Correct
Note de 0,50 sur 0,50
Marquer la question

En Python, dans une structure de répétition, les instructions Break et Continue sont équivalentes.

Veuillez choisir une réponse.
Vrai
Faux ✔

La réponse correcte est « Faux ».

Question 6
Correct
Note de 0,50 sur 0,50
Marquer la question

La comparaison entre deux chaines de caractères compare les codes ASCII des deux premiers éléments, s'ils diffèrent cela détermine le résultat de la comparaison, sinon les deux éléments suivants sont comparés, et ainsi de suite, jusqu'à ce que l'une des deux chaines soit épuisée.

Veuillez choisir une réponse.
Vrai ✔
Faux

La réponse correcte est « Vrai ».

Question 7
Correct
Note de 0,50 sur 0,50
Marquer la question

En Python, il existe plusieurs types. Certains sont mutables d'autres sont immutables. Sélectionner tous les types immutables.

a. str ✔
b. list
c. set
d. tuple ✔
e. dict
f. int ✔

Votre réponse est correcte.
Les réponses correctes sont :
str,
int,
tuple

Question 8
Correct
Note de 1,00 sur 1,00
Marquer la question

Soit la fonction suivante :

```
1. def some_function(some_list):
2.     some_list = sorted(some_list)
3.     compteur = 0
4.     while some_list:
5.         some_list = some_list.pop()
6.         compteur += 2
7.
8.     return len(some_list), compteur

>>> print(some_function(["a", "b", "d", "c"]))
```

Quel résultat obtient-on?

a. (0, 0)
b. (4, -2)
c. "AttributeError: 'str' object has no attribute 'pop'" ✔

Votre réponse est correcte.
La réponse correcte est :
"AttributeError: 'str' object has no attribute 'pop'"

Question 9
Correct
Note de 1,00 sur 1,00
Marquer la question

Soit la fonction suivante:

```
1. def diviser(numerateur, denominateur):
2.     somme = 0
3.     for i in range(len(numerateur)):
4.         somme += denominateur[0] / numerateur[i]
5.
6.     return somme

5) on appelle cette fonction ainsi: diviser([2.5, 8.5, 10], [3.5, 4.2
```

Réponse : 1.0 ✔

La réponse correcte est : 1

Question 10
Terminé
Noté sur 1,00
Marquer la question

Expliquez brièvement pourquoi il n'est pas possible d'utiliser des listes comme clefs dans un dictionnaire.

Les clés d'un dictionnaire doivent être immutables, car elles sont hashées pour correspondre à la référence. Une liste est une classe qui ne peut être hashée, car elle est mutable. Si nous pouvions utiliser une liste comme clé, il ne serait pas certain que nous nous référencions à cet une comparaison des éléments de la liste) de vérifier avec la référence exacte.

Question 11
Correct
Note de 1,00 sur 1,00
Marquer la question

Les années bissextiles comportent un jour supplémentaire. Cette irrégularité permet de conserver l'écart entre la durée d'une année calendaire et celle d'une année solaire.
Une année sera bissextile si :

- Elle est divisible par 4 (soit une année bissextile tous les 4 ans)
- Elle n'est pas divisible par 100 (2100 n'est pas bissextile, sauf cas suivant
- Elle est divisible par 400 (2000 est bissextile, car divisible par 400 - donc par 4 et 100, mais la règle du 400 est plus forte)

Compléter ce programme, qui doit déterminer si une année est bissextile ou non. On considère que l'année est un nombre entier strictement positif. Le programme doit retourner True dans ce cas et l'année est bissextile et False sinon.

**Par exemple:**

| Test | Résultat |
|---|---|
| `print(est_bissextile(1999))` | False |

**Réponse :** (régime de pénalités : 0 %)

Réinitialiser la réponse

```
def est_bissextile(annee: int) -> bool:
    if annee % 4 == 0:
        if not annee % 100 == 0:
            return True

        if annee % 400 == 0:
            return True

    return False
```

| Test | Résultat attendu | Résultat obtenu | |
|---|---|---|---|
| `print(est_bissextile(1999))` | False | False | ✔ |
| `print(est_bissextile(2400))` | True | True | ✔ |
| `print(est_bissextile(2100))` | False | False | ✔ |
| `print(est_bissextile(2020))` | True | True | ✔ |

Tous les tests ont été réussis ! ✔

**Solution de l'auteur de la question (Python3):**

```
def est_bissextile(annee):
    # On suppose que annee > 0
    return annee % 4 == 0 and annee % 100 != 0 or annee
```

Correct
Note pour cet envoi : 1,00/1,00.

Question 12
Correct
Note de 4,00 sur 4,00
Marquer la question

Lorsque l'on dispose d'un nombre important de données, leur stockage devient un défi. Il faut trouver un moyen de stocker les données en prenant le moins de place possible.

Soit un système qui relève la température de l'air au cours de la journée, avec une résolution de 0,5 °C. La fréquence est de une mesure par minute. Sur une année, on obtient environ 500 000 mesures.

Les valeurs sont plutôt stationnaires ; la température varie lentement. Par conséquent, les tableaux de données obtenus comportent des séries de valeurs identiques.

Pour compresser des données, une méthode consiste à remplacer une série de n fois la valeur x par le couple (x, n).

Par exemple, [12, 21, 13, 13, 14, 3] représente [12, 12, 13, 14, 14, 14].

Compléter le code suivant pour qu'il compresse la liste de valeurs donnees.

**Par exemple:**

| Test | Résultat |
|---|---|
| `donnees = [21, 21, 22, 23, 23] compresse(donnees) print(donnees)` | `[21, 2], [22, 1], [23, 2]]` |

**Réponse :** (régime de pénalités : 0 %)

Réinitialiser la réponse

```
def compresse(donnees: list) -> list:
    if len(donnees) == 0:  # TODO: Cas d'une liste vid
        return  # TODO: On arrête la fonction

    valeur = donnees[0]
    position = 0
    compteur = 1
    while True:
        if 1 == len(donnees):  # TODO: Si on est arriv
            donnees.insert(position + 1,
                           [valeur, compteur])
            return donnees  # TODO: Arrêt de la boucle
        elif valeur == donnees[position + 1]:  # TODO: Si on trouve
            compteur += 1  # TODO: On incrémente le co
        else:  # TODO: Si on trouve une autre valeur
            del donnees[position : position + compteur
            donnees.insert(position, [valeur, compteur
            compteur = 1
            position += 1
            valeur = donnees[position]
```

| Test | Résultat attendu | Ré |
|---|---|---|
| `donnees = [] compresse(donnees) print(donnees)` | `[]` | `[]` |
| `donnees = [21, 21, 22, 23, 23] compresse(donnees) print(donnees)` | `[[21, 2], [22, 1], [23, 2]]` | `[[` |
| `donnees = [12, 12, 13, 14] compresse(donnees) print(donnees)` | `[[12, 2], [13, 1], [14, 1]]` | `[[` |

Tous les tests ont été réussis ! ✔

**Solution de l'auteur de la question (Python3):**

```
def compresse(donnees):
    if len(donnees) == 0:
        return
    else:
        valeur = donnees[0]
        position = 0
        compteur = 1
        while True:
            if 1 == len(donnees):
                donnees.append([valeur, compteur])
                return donnees
            elif valeur == donnees[position + 1]:
                compteur += 1
            else:
                del donnees[position : ]
                donnees.insert(position, [valeur, compteur
                compteur = 1
                position += 1
                valeur = donnees[position]
```

Correct
Note pour cet envoi : 4,00/4,00.