

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique**IN2T Informatique***Examen Janvier 2016***Consignes générales**

- Pas de calculatrice.
- Feuilles de brouillon incluses, à rendre avec sa copie.
- Réponses au bic ou stylo, noir ou bleu.

Mise en forme du code

- Vous pouvez ajouter des commentaires pour expliquer des parties non triviales de vos réponses.
- Faites attention au soin de vos réponses, et en particulier à rendre l'indentation explicite.
- Ne pas recopier les entêtes des fonctions déjà fournies.

Réservé aux correcteurs (questionnaire A)

Q1						/3
Q2						/4
Q3						/3
Q4						/5
Q5						/5
Total						/20

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique

Aide-mémoire

Conditions

```
a = -5
if a == 0:
    print("a est nul")
elif a < 0:
    print("a est négatif")
else:
    print("a est positif")
```

Opérations	Descriptions
< et >	strictement inférieur ou supérieur
<= et >=	« inférieur ou égal » ou « supérieur ou égal »
= et !=	égal ou différent
and et or	« et » logique ou « ou » logique
not	« non » logique
in	vérifie qu'une valeur appartient à une séquence

Boucles

```
i = 0
while i < 5:
    i += 1
```

```
for variable in séquence:
    bloc d'instructions
```

Fonctions

```
# Définition de la fonction
def nomDeLaFonction(paramètres):
    bloc d'instructions

# Appel de la fonction
nomDeLaFonction(valeurs des paramètres)
```

Pour définir une valeur de retour dans le bloc d'instructions, on utilise le mot-clé `return`

Collections

```
s = '' # String vide
l = [] # Liste vide
t = () # Tuple vide
e = set() # Ensemble vide
d = {} # Dictionnaire vide

s[0] # Premier élément "s" (aussi valable pour l et t)
d['key'] # Valeur associée à la clé "key" dans "d"
len(c) # Taille de la collections "c"

s[0:2] # Nouvelle chaine des indices 0 compris
# à 2 non-compris (aussi valable pour l et t)

del(l[1]) # Supprime l'élément d'indice 1
del(d['key']) # Supprime l'élément de clé "key"

l[2:3] = ['A', 'B'] # Remplace une tranche de la liste "l"
l.append('après') # Ajout un élément en fin de "l"

l.pop(3) # Supprime et renvoie le 4e élément de "l"
l.insert(2, 'A') # Insère un élément à l'indice 2 dans "l"

e.add('bla') # ajoute 'bla' à l'ensemble "e"
e.remove('bla') # supprime 'bla' de l'ensemble "e"

s.rstrip() # supprime les caractères blancs en fin "s"
s.split(',') # découpe le string selon les ','
```

Opérations	Descriptions
+	Concatène deux séquences
« entier » *	Répète une séquence
« ensemble » - « ensemble »	Fait la différence de 2 ensembles
« ensemble » « ensemble »	Fait l'union de 2 ensembles
« ensemble » & « ensemble »	Fait l'intersection de 2 ensembles

Classes

```
class Quantity:
    # Constructeur
    def __init__(self, value):
        self._value = value

    # Accesseur
    @property
    def value(self):
        return self._value

    # Mutateur
    @value.setter
    def value(self, value):
        self._value = value

    # Représentation
    def __str__(self):
        return str(self.value)

    # Surcharge de la somme
    def __add__(self, other):
        return Quantity(self.value + other.value)
```

D'autres surcharges sont possible avec `__sub__`, `__mul__`, `__div__`, `__lt__`, `__le__`, `__eq__`, `__gt__`, `__ge__`

Gestion d'erreurs

```
# Arrêt du programme si "x" n'est pas entier
assert type(x) == int, 'x must be int'

# Récupération et affichage d'une exception
try:
    a = int(input('a?: '))
except Exception as e:
    print(e)

# Définir un type d'exception
class MyException(Exception):
    pass

# Lancement d'une exception
raise MyException()
```

Quelques types d'exceptions utiles : `ValueError`, `ZeroDivisionError`, `IOError`, `FileNotFoundError`

Fichier

```
# Lecture d'un fichier texte
with open('data.txt') as file:
    print(file.read())

# Écriture d'un fichier texte
with open('data.txt', 'w') as file:
    file.write('hello world !!')
```

La méthode `readlines()` d'un fichier ouvert en lecture permet de récupérer la liste de toutes les lignes du fichier.

Une boucle `for .. in ..` sur un fichier en lecture permet d'itérer sur les lignes du fichier.

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique**1 Vrai ou faux (3 points)**

(Connaissances générales)

Pour chacune des affirmations suivantes, indiquez si elle est vraie ou fausse. *Si vous avez répondu correctement à toutes les affirmations d'un bloc, vous obtenez un point, si vous avez fait une faute vous obtenez un demi-point et sinon vous avez zéro (une abstention compte comme une faute).*

1.1 Structure de donnée

Affirmation	Vrai	Faux
A/ Un dictionnaire peut être considéré comme un ensemble.	<input type="checkbox"/>	<input type="checkbox"/>
B/ Un ensemble peut contenir des éléments de différents types.	<input type="checkbox"/>	<input type="checkbox"/>
C/ Le dictionnaire déclaré par l'instruction suivante contient trois éléments. <code>dico = {'apple': 12, 'banana': 7, 'apple': 8}</code>	<input type="checkbox"/>	<input type="checkbox"/>
D/ On peut ajouter un élément à un tuple déjà existant à l'aide de la méthode <code>append</code> .	<input type="checkbox"/>	<input type="checkbox"/>
E/ L'ensemble <code>s</code> initialisé par l'instruction suivante contient un seul élément. <code>s = {e for e in range(10) if 1 % e == 0}</code>	<input type="checkbox"/>	<input type="checkbox"/>

1.2 Programmation orientée objet

Affirmation	Vrai	Faux
A/ Un attribut d'un objet n'est accessible qu'en lecture.	<input type="checkbox"/>	<input type="checkbox"/>
B/ Une condition est une expression de type booléen.	<input type="checkbox"/>	<input type="checkbox"/>
C/ L'exécution des instructions suivantes affiche (8, 1) à l'écran. <pre>class Foo: def __init__(self, x, y): self.x = x self.y = y x = Foo(8, 1) print(x)</pre>	<input type="checkbox"/>	<input type="checkbox"/>
D/ Une classe peut contenir plusieurs constructeurs (différentes définitions).	<input type="checkbox"/>	<input type="checkbox"/>
E/ Une exception est représentée par un objet.	<input type="checkbox"/>	<input type="checkbox"/>

Nom : 15/01/2016 13:30
Matricule : (105 minutes)

B2010 — Informatique**1.3 Concepts informatiques**

Affirmation	Vrai	Faux
A/ Les deux expressions booléennes suivantes sont équivalentes : — not (rainy or windy) — (not rainy) or (not windy)	<input type="checkbox"/>	<input type="checkbox"/>
B/ La traduction d'un nom de domaine en l'adresse IP correspondante se fait par le serveur DHCP.	<input type="checkbox"/>	<input type="checkbox"/>
C/ HTML est un langage de programmation spécialisé pour créer des documents en PDF.	<input type="checkbox"/>	<input type="checkbox"/>
D/ Paul Otlet est un belge.	<input type="checkbox"/>	<input type="checkbox"/>
E/ Les bornes WiFi que les particuliers possèdent chez eux, reliées à la ligne téléphonique ou de télévision, sont à la fois un modem et un routeur.	<input type="checkbox"/>	<input type="checkbox"/>

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique**2 Théorie (4 points)**

(Compréhension d'un concept informatique)

Définir les notions d'**objet et de classe**, ainsi que le lien qui existe entre les deux. Illustrer ensuite votre définition sur base d'un **exemple simple**.

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique*(suite de la réponse)*

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique**3 Combien de lignes valides ? (3 points)**

(Niveau élémentaire)

Définir une fonction qui prend le nom d'un fichier texte en paramètre et qui renvoie le nombre de lignes valides contenues dans ce fichier ; une ligne étant valide si elle contient un nombre entier. Si le fichier n'existe pas, où si une erreur se produit pendant la lecture du fichier, la fonction doit renvoyer -1 .

```
def countlines(path) :
```

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique*(suite de la réponse)*

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique**4 Backup de SMSs (5 points)**

(Niveau moyen)

Supposons que les SMSs de votre téléphone soient stockés sous la forme d'une liste de dictionnaires. Chaque dictionnaire stocke l'expéditeur ou le destinataire selon que ce soit un SMS entrant ou sortant, la date et le message. Voici un exemple d'une telle liste :

```
SMSs = [  
  {"type" : "out", "contact" : "Alexis", "message" : "Salut Alexis, ça va ? :-)", "date" : "2015-12-05 08:30"},  
  {"type" : "in", "contact" : "Marie", "message" : "Hey ! T'es mort ?", "date" : "2015-12-07 15:40"},  
  {"type" : "in", "contact" : "Alexis", "message" : "ouais...", "date" : "2015-12-05 09:22"}  
]
```

Définir une fonction qui prend en paramètre une liste de SMSs et qui backupe les SMSs dans des fichiers texte. Un fichier est créé par contact et contient respectivement le type, la date et le message, séparés avec le caractère ¶. Par exemple, le fichier Alexis doit contenir les deux lignes suivantes :

```
out¶2015-12-05 08:30¶Salut Alexis, ça va ? :-)  
in¶2015-12-05 09:22¶ouais...
```

def export(SMSs) :

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique*(suite de la réponse)*

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique**5 Le plus populaire (5 points)**

(Niveau avancé)

Soit les préférences d'un ensemble de personnes concernant des plats, représentées par un nombre entier sur une échelle de 0 (n'aime pas) à 5 (adore), stockées dans un dictionnaire. Ce dernier est structuré comme suit : les clés représentent les personnes et les valeurs sont une liste de votes ; un vote étant un dictionnaire contenant comme clé le nom du plat et comme valeur la note attribuée. Voici un exemple d'un tel dictionnaire :

```
prefs = {"Lurkin" : [{"Nem" : 4}, {"Frites" : 4}], "Marchand" : [{"Durum" : 3}, {"Nem" : 2},  
{"Chicon" : 1}], "Mélotte" : [], "Combéfis" : [{"Nem" : 4}, {"Patte de poulet" : 4}]}
```

Définir une fonction qui prend en paramètre un tel dictionnaire de préférences et qui renvoie l'ensemble des plats qui ont la plus grande note de préférence. Pour l'exemple ci-dessus, la fonction doit renvoyer l'ensemble : {"Nem", "Frites", "Patte de poulet"}.

```
def preferredmeal(prefs) :
```

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique*(suite de la réponse)*

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique

Brouillon

Nom :

15/01/2016 13:30

Matricule :

(105 minutes)

B2010 — Informatique*(brouillon suite)*