Université Bejaia / Fac Sciences exactes /dpt. MI / L2 AP

20

RATTRAPAGE Langage Evolué | Nom : …… **Prénom :** ……Groupe : ……

EXERCICE 1 : QCM (10 points)

Q1 : Que va renvoyer la commande suivante :

>>> type ([(1,2)])

- ✓ **<class 'list'>**
- ☐ <class 'tuple'>
- ☐ <class 'dict'>
- ☐ <class 'int'>

Q2 : Que va renvoyer la commande suivante :

>>> type (len({ 1:2 , 4:8 }))

- ☐ <class 'list'>
- ☐ <class 'tuple'>
- ☐ <class 'dict'>
- ✓ **<class 'int'>**

Q3 : Soit :

```
>>> a = [ (1 , 2 , 3, "Math") , { 1:2 } , [ 1 , 2 ] ]
>>> len(a)
```

Que va afficher le code suivant : **len(a)**

3

Q4 : Que va afficher le code suivant :

```
x = True
y = False
z = True
if x or (y and z):
    print ("OUI")
else:
    print ("NON")
```

OUI

Q5 : Indiquez la liste des valeurs qui seront affichées par le code suivant :

```
liste=(1,2,3,(4,5))
for x in  range(len(liste)):
    print(x)
```

- ☐ 2, 4, 6
- ☐ 0,1, 4, 5
- ✓ **0, 1, 2, 3**
- ☐ 0, 1, 4, 5, 6, 7, 8, 9
- ☐ 1, 2, 4, 5, 6

Q6 : Indiquez la liste des valeurs qui seront affichées par le code suivant :

```
liste=(1,2,3,(4,5))
for x in  range(1, len(liste), 3):
    print(x)
```

1

Q7: Indiquez ce que va afficher le code suivant :

```
try:
    print (2+len("Math-STID"))
except:
    print ("erreur")
else:
    print ("OK")
```

11
OK

Q8 : Indiquez ce que va afficher le code suivant si l'utilisateur tape 8 en réponse à l'instruction « input »:

```
try:
    a = input("Donne une valeur ")
except:
    print ("erreur")
else:
    print ("OK")
```

OK

Q9 : Soit une liste initialisée comme suit :

>>> **Villes = ['Bejaia', 'Alger', 'Oran', 'Jijel']**

Quelle est la commande python permettant de créer une chaine « S » composée des éléments de la liste « modules » séparés par le caractère « **/** ».

- ☐ **S = join(Villes , "/")**
- ☐ **S = "/".join(Villes)**
- ☐ **S = Villes + "/"**
- ☐ **S = Villes.split("/")**

Q10 : Soit les commandes suivantes :

```
>>> pays = {}
>>> pays['Bejaia']='algérie'
>>> pays['bejaia']='algérie'
>>> pays['Alger']='algérie'
>>> pays['Alger']='Algérie'
>>> print(len(pays))
```

Que va afficher la dernière commande ?

3

Q11 : Soit les commandes suivantes :

```
>>> eleve = ['amal', 'said', 'mohand', 'amira']
>>> A = eleve
>>> A.append(10)
>>> print(len(eleve))
```

Que va afficher la dernière commande ?

5

Q12 : Soit les commandes suivantes :

```
>>> fruits = ['pomme', 'tomate', 'abricot']
>>> print ( fruits[-2] )
```

Que va afficher la dernière commande ?

tomate

Q13 : Soit les commandes suivantes :

```
>>> fruits = ['pomme', 'tomate', 'abricot']
>>> print ( fruits[-1][-3] )
```

Que va afficher la dernière commande ?

c

Q14 : Soit les commandes suivantes :

```
>>> fruits = ['pomme', 'tomate', 'abricot']
>>> print (fruits[0][0:3] )
```

Que va afficher la dernière commande ?

pom

Q15 : Que va afficher le programme suivant :

```
>>> s = "Bejaia Alger : Oran"
>>> r = s.split(" : ")
>>> r.append(10)
>>> print(r[1])
```

Oran

Q16 : Que va afficher le programme suivant :

```
>>> def f(x, y=2, z=5):
        return x+y+z
>>> print(f(z="2", x="BAC", y="3"  ))
```

BAC32

Q17 : Que va afficher le programme suivant :

```
>>> def f(x, y=2, z=5):
        return x+y+z
>>> print(f(10))
```

17

Q18 : Que va afficher le programme suivant :

```
>>> def f(x, y=10, z="14"):
        return x+y+z
>>> print(f("Physique","Math"))
```

'PhysiqueMath14'

Q19 : Que va afficher le programme suivant :

```
>>> equipes = ('JSK', 'MCA', 'MOB', 'JSMB', 'RSK')
>>> f = open('equipes.txt', 'w')
>>> for e in equipes:
        f.write(e+"\n")
>>> f.close()
>>> f = open('equipes.txt', 'r')
>>> texte = f.read().split("\n")
>>> print(texte[2])
```

MOB

Q20 : Que va afficher le programme suivant :

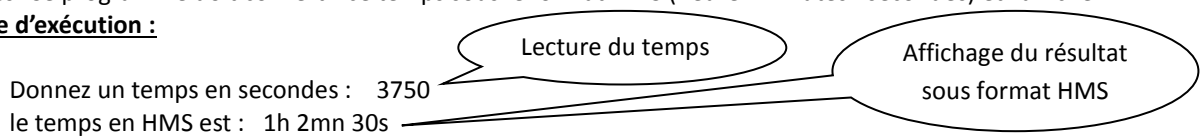
```
>>> equipes = ('JSK', 'MCA', 'MOB', 'JSMB', 'RSK')
>>> f = open('equipes.txt', 'w')
>>> for e in equipes:
        f.write(e+"\n")
>>> f.close()
>>> f = open(' equipes.txt', 'r')
>>> print(f.read().strip('\n').split("\n"))
```

['JSK', 'MCA', 'MOB', 'JSMB', 'RSK']

Nom : …… **Prénom :** ……Groupe : ……

EXERCICE 2 (5 points) : Ecrire un programme qui lit (**une seule fois**) une valeur représentant un temps exprimé en secondes. Ce programme doit convertir ce temps sous le format HMS (heure : minutes : secondes) et l'afficher.

Exemple d'exécution :



Indication importante : Vous devez impérativement écrire trois fonctions permettant chacune de donner le nombre d'heures, de minutes et de secondes. Vous devez aussi écrire une fonction « ***lireTemps***() » permettant de renvoyer un entier lu à partir du clavier. Attention, cette fonction doit effectuer un contrôle de validité de la valeur saisie !

Exemple : ***heures(3750)*** revoie la valeur 1 pour dire une heure.
minutes(3750) revoie la valeur 2 pour dire deux minutes (3750 – 1*3600 = 150, ce qui donne 2 minutes)
secondes(3750) revoie la valeur 30 pour dire 30 secondes (3750 – 1*3600-2*60) = 30
lireTemps() permet de renvoyer un entier lu au clavier tout en s'assurant de la validité de la saisie.

```
def heures(temps):
    return int(temps/3600)

def minutes(temps):
    return int((temps - heures(temps)*3600)/60)

def secondes(temps):
    return temps - heures(temps)*3600 - minutes(temps)*60

def lireTemps():
    while True:
        try:
            t = int(input("donnez un temps en secondes : "))
            return t
        except :
            print("vous devez saisir un entier positif ou nul!")

# Ici débute le programme principal
t = lireTemps()
h = str(heures(t))+ "h "
m = str(minutes(t))+"mn "
s = str(secondes(t))+"s"
print("Le temps que vous avez sasie correspond à ceci : "+ h + m +s)
```

EXERCICE 3 (5 points) : Ecrire un programme qui lit un fichier de données nommé « **pays.txt** ».
Ce fichier contient une liste de pays avec leurs villes telle montré ici →

```
# Algérie
Bejaia
Alger
Oran

# France
Paris
Marseille
Lyon

# Tunisie
Sousse
Carthage
Tunis
```

On vous demande d'écrire un programme qui

1. Ouvre ce fichier en mode lecture
2. lit ce fichier dans une variable nommé « ***paysVilles*** ». Cette variable doit être une liste qui contiendra chacune des lignes du fichier (des pays et des villes). Il faut éliminer les lignes vides !
3. Crée un dictionnaire nommé « ***villes*** ». Ce dictionnaire doit avoir comme clés les noms des pays et comme valeurs des listes des villes.
4. Construit le dictionnaire « ***villes*** » à partir de la liste « ***paysVilles*** »
5. Affiche la liste des villes d'un pays saisie au clavier
6. Affiche le pays d'une ville saisie au clavier

Exemple d'exécution :

```
Donnez un pays, je vous donne ses villes : Algérie
Les villes de Algérie sont : Bejaia, Alger, Oran
Donnez une ville, je vous donne le pays où elle se trouve : Tunis
Le pays où se trouve Tunis est Tunisie
>>>
```

```
f = open("pays.txt", "r")                                   0.5  Ouverture du fichier

paysVilles = f.read()                                       0.5  Lecture du fichier
paysVilles = paysVilles.split("\n")

while '' in paysVilles:                                     0.5  Elimination de la dernière ligne vide
   paysVilles.remove('')


villes = {}
for i in range(len(paysVilles)):
   if "#" in paysVilles[i]:
      v=paysVilles[i].strip("#").strip(" ")                     Construction du dictionnaire villes   0.5
      villes[v]=[]
   else:
      villes[v].append(paysVilles[i])

p = input("donnez moi un pays, je vous donne ses villes : ")   0.5  Lecture d'un pays au clavier

while p not in villes.keys():
   print("\t ce pays n'existe pas dans mon fichier!")
   p = input("\ndonnez moi un pays, je vous donne ses villes : ")   Vérification de l'existence du pays   0.5

for v in villes[p]:                                         0.5  Affichage de la liste des villes d'un pays donné au clavier
   print(v)

v = input("\ndonnez moi une ville, je vous donne la pays où elle se trouve : ")
                                                            0.5  Lecture d'une ville au clavier

trouve = False
for p in villes.keys():
   if v in villes[p]:                                       0.5  Recherche du pays
      trouve = True
      print(p)

if not trouve:                                                    Vérification de l'existence du pays   0.5
   print ("\tDésolé, je n'ai pas trouvé de pays correspondant à votre ville")
```