

Rattrapage Langage Evolué | Nom : Prénom : Groupe :

EXERCICE 1 : QCM (11 points)

Q1 : Que va renvoyer la commande suivante :

>>> type (" {1,2} ")

- <class 'list'>
- <class 'tuple'>
- <class 'dict'>
- <class 'str'>

Q2 : Que va renvoyer la commande suivante :

>>> type ([("1:2" , "4:8") , (1,4)])

- <class 'list'>
- <class 'tuple'>
- <class 'dict'>
- <class 'str'>

Q3 : Soit :

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

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

Q4 : L'expression

"fruits" in ("kiwi","banane","pomme") retourne:

- <class 'tuple'>
- True
- False
- Str

Q5 : Les séquences immuables sont...:

- Chaine
- Tuple
- Liste
- Dictionnaire

Q6 : Soit la variable fruits initialisée avec le **tuple** ("kiwi","pomme","mangue"). L'instruction fruits[0] = "poire" est elle correcte ? :

- Oui
- Non
- Oui sous certaines conditions

Q7: Soit la variable fruits initialisée avec le **tuple** ("kiwi", "pomme", "mangue", "poire", "orange", "figue").

L'instruction fruits[1:3] retourne... :

- ()
- ("pomme","poire")
- ("pomme","mangue")
- ("pomme","mangue","poire")

Q8 : Indiquez ce que va afficher le code suivant si l'utilisateur tape 5 en réponse à l'instruction « input »:

```
try:
    x = float( input("Donne une valeur") )
except:
    print ("Erreur")
else:
    print (x)
```

Q9 : Soit une liste initialisée comme suit :

```
>>> Mots = ['Bejaia', 'est', 'une', 'ville.']}
```

Quelle est la commande python permettant de créer, à partir de la variable « **Mots** », une nouvelle variable « **phrase** » contenant une chaîne composée des éléments de la variable « **listeMots** » séparés par des espaces.

Q10 : Soit les commandes suivantes :

```
>>> pays = { }
>>> pays["Europe"] = [ "Italie", "Espagne", "Turquie"]
>>> pays["Europe"] = [ "France", "Belgique", "Italie"]
>>> pays["Europe"] = [ "Allemagne", "Turquie"]
>>> print(len(pays))
```

Que va afficher la dernière commande ?

Q11 : Que va afficher le code suivant ?:

```
def afficher():
    print ( " Bejaia " )
    print (" Algérie ")
```

Nom : Prénom : Groupe :

Q12 : Soit les commandes suivantes :

```
>>> Club = ['JSK', 'MOB', 'JSMB']
>>> print ( Club[-1] )
```

Que va afficher la dernière commande ?

Q13 : Que va afficher le code suivant ?:

```
def afficher():
    print ( " Salut " )
    return
    print ( " Mes amis " )
afficher()
```

Q14 : Que va afficher le code suivant ?:

```
>>> x = "10"
>>> y = 20
>>> print( float( str( int(x)+y )*2 ) )
```

Q15 : Que va afficher le programme suivant :

```
>>> s = "*****Béjaia ????"
>>> r = s.strip( "*" )
>>> r = r.strip( " ?" )
>>> print (r)
```

Q16 : Que va afficher le programme suivant :

```
>>> v = 10
>>> def f ( x , y = 1):
    x = x - y
    return x+v
>>> print ( f ( v ) )
```

Q17 : Que va afficher le programme suivant :

```
>>> v = 10
>>> def f ( x , y = 1):
    x = x - y
    return x+v
>>> print ( f ( 5 , 6 ) )
```

Q18 : Que va afficher le programme suivant :

```
>>> v = 6
>>> def f ( x , y = 1):
    x = x + y
    return x + str ( v )
>>> print ( f ( " Ville " , " Bejaia " ) )
```

Q19 : Que va afficher le programme suivant :

```
>>> D = dict()
>>> def f(x):
    For i in range(4):
        D[i] = i*2
    >>> f(10)
    >>> for v in D.values():
        print(v)
```

Q20 : Que va afficher le programme suivant :

```
>>> def f ( x , y ):
    return x + y
>>> print ( f ( [1, 2, 3], [0] ) )
```

Q21 : Que va afficher le programme suivant :

```
>>> f = open("test.txt", "w")
>>> f.write("Bonjour Monsieur")
>>> f.close()
>>> f=open("test.txt", "r")
>>> texte = f.read()
>>> print(texte[0:7])
```

Q22 : Que va afficher le programme suivant :

```
>>> f = open("test.txt", "w")
>>> f.write("Bonjour Monsieur")
>>> f.close()
>>> f=open("test.txt", "r")
>>> texte = f.read()
>>> texte = f.read()
>>> print(texte+"1")
```

Nom : Prénom : Groupe :

EXERCICE 2 (4 points) : Écrire une fonction "*occurencesMots*" ayant comme argument une chaîne de caractères représentant une phrase et qui renvoie un dictionnaire contenant la fréquence (nombre d'apparitions) de tous les mots de la chaîne entrée.

Exemple d'exécution :

```
>>> phrase = "je suis arrivée à Bejaia à 16 heures"
>>> occurencesMots(phrase)
{ 'je': 1 , 'suis': 1 , '16': 1 , 'Bejaia': 1 , 'heures': 1 , 'à': 2 , 'arrivée': 1 }
```

Indication : On suppose que la phase est correctement construite.

```
def occurencesMot(phrase):
```

```
    occurences = {}
```

```
    caractèresPonctuation = [ " " , ";" , ":" , ".", "?", "!", "'", ":" ]
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

```
    .....
```

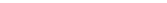
```
    .....
```

```
    .....
```

```
    .....
```

```
    return occurences
```

Nom : Prénom : Groupe :

EXERCICE 3 (5 points) : On considère un fichier « recettes.txt » de recette de cuisine indiqué ici. 

Chaque ligne du fichier « recettes.txt » contient le nom d'un plat suivi de « : » suivi d'une liste d'ingrédients séparés par des virgules.

On vous demande d'écrire un programme qui permet de donner la liste des plats (indiqués dans le fichier « recettes.txt ») qui utilisent un ingrédient saisi au clavier.



Pizza : sel, tomate, huile, patte à pizza

Omlette au fromage : oeuf, fromage, huile, sel

Tarte au citron : sucre, citron, patte à tarte

Exemple d'exécution

donnez un ingrédient : sel

voici la liste des plats utilisant l'ingrédient : sel

-->pizza

-->omlette au fromage

Bon courage

Exercice 2 :

```
>>> def compterMots(phrase):
    occurrences = {}
    caractèresPonctuation = [" ", ",", ";", ".", "?", "!", "", ":"]
    listeMots = []
    mot = ""
    for c in phrase:
        if c in caractèresPonctuation:
            listeMots.append(mot)
            mot = ""
        else :
            mot = mot + c
    if mot != "":
        listeMots.append(mot)

    for m in listeMots:
        if m in occurrences.keys():
            occurrences[m]=occurrences[m]+1
        else:
            occurrences[m]=1

    return occurrences
```

Exercice 3

```
f = open("recette.txt", "r")
texte = f.read().strip().split("\n")
recette = {}
for e in texte:
    platIngrédients = e.split(":")
    plat = platIngrédients[0]
    ingrédients = platIngrédients[1].split(",")
    for i in range(len(ingrédients)):
        ingrédients[i] = ingrédients[i].strip()
    recette[plat] = ingrédients

ing = input("donnez un ingrédient ")
print("\nvoici la liste des plats utilisant l'ingrédient : ", ing)
trouve = False
for p, i in recette.items():
    if ing in i:
        print("\t-->"+p)
        trouve = True
if not trouve:
    print("je n'ai pas trouvé de plat utilisant votre ingrédient")
```