

Exercice 1 : Questions à réponses courtes (12 points)

Indication : n'oubliez pas d'indiquer dans vos réponses les guillemets pour les chaînes de caractères, les crochets pour les listes et les accolades pour les dictionnaires. Dans le cas contraire, votre réponse sera considérée comme fautive !

☞ Indiquez ce que vont afficher les instructions suivantes (on suppose qu'on utilise Python 3.4):

<p>Q1 : print("1250".isnumeric()) : True</p> <p>Q2 : print("Math".isdigit()) : False</p> <p>Q3 : print("12.50".endswith("12")) : False</p> <p>Q4 : print (type ([{ 1 : 12 }])) : <class 'list'></p> <p>Q5 : print (type ((,))) : <class 'tuple'></p> <p>Q6 : print (type (int ("12 "))) : <class 'int'></p>	<p>Q7 : print (list (range (1 , 10 , 2)) [0]) : 1</p> <p>Q8 : print (list (range (1 , 10 , 2)) [-1]) : 9</p> <p>Q9 : print ("*"12*Math*" .strip("*")) : 12*Math</p> <p>Q10 : print ("+" .join (["1" , "2" , "3"])) : 1+2+3</p> <p>Q11 : print("12"*2 + "20") : 121220</p> <p>Q12 : print("STID 2015".replace("2015", "2016")) : STID 2016</p>
---	--

☞ Soit la liste **notes = ["A" , "C" , "B"]**, indiquez ce que va contenir la liste **notes** pour chacune des instructions suivantes (attention, ces instructions sont exécutées séquentiellement « l'une derrière l'autre ! »):

Q13 : notes.reverse() : ['B', 'C', 'A']	Q14 : notes.sort() : ['A', 'B', 'C']
Q15 : notes.pop(-1) : ['A', 'B']	Q16 : notes = notes + [14] : ['A', 'B', 14]

☞ Indiquez ce que vont afficher les instructions suivantes :

<p>Q17</p> <pre>notes = list (range (2 , 10)) res = list() for e in notes : if e%3 ==0 : res.append(e) print (res)</pre> <p>Réponse : [3, 6, 9]</p>	<p>Q18</p> <pre>notes = ["2", "7", "C"] res = list() for a in notes : if a.isalpha () : res.append(a*2) print (res)</pre> <p>Réponse : ['CC']</p>
<p>Q19</p> <pre>notes = ["2" + "0" , 7 , 4] print (notes.index(7))</pre> <p>Réponse : 1</p>	<p>Q20</p> <pre>notes = ["4" + "1" , 4] print (notes [0])</pre> <p>Réponse : 41</p>

☞ Soit le code python suivant :

```
def f ( x = [0] , y = 0 , z = [ 10 ] ) :
    return x+[y]+ z
```

Indiquez ce que va contenir la variable **RES** à l'issue de l'exécution des instructions suivantes :

<p>Q21 : RES = f () : [0, 0, 10]</p> <p>Q22 : RES = f ([1]) : [1, 0, 10]</p> <p>Q23 : RES = f ([1] , 2) : [1, 2, 10]</p> <p>Q24 : RES = f ([1] , 2 , list (range (1))) : [1, 2, 0]</p>
--

Exercice 2 : Ecriture d'une fonction simple (2 points)

Ecrire une fonction nommée **admis()** qui retourne la liste des étudiants admis sachant qu'elle a pour argument un dictionnaire nommé **moyenneDe** ayant pour clé le nom de l'étudiant et comme valeur sa moyenne. Un étudiant sera considéré comme admis si sa moyenne est supérieure ou égale à 10

```
def admis(moyenneDe) :  
    elevesAdmis = list()  
    for e, m in moyenneDe.items():  
        if m >= 10:  
            elevesAdmis.append(e)  
    return elevesAdmis
```

Exercice 3 : Ecriture d'une fonction simple (2 points)

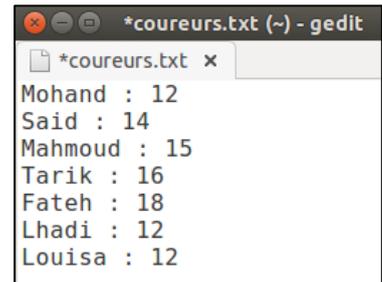
Ecrire une fonction nommée « **identiques ()** » qui compare deux fichiers (on suppose que ce sont des fichiers « texte »). Cette fonction aura en entrée 3 paramètres : le nom du premier fichier, le nom de second fichier et le 3^{ème} paramètre est le dossier où se trouvent les deux fichiers. En sortie, elle renvoi la valeur **True** si les deux fichiers sont identiques ou **False** sinon.

```
def identiques(fichier1, fichier2, dossier) :  
    f1 = open( dossier + "/" + fichier1 , "r")  
    texte1 = f1.read()  
    f1.close()  
    f2 = open( dossier + "/" + fichier2 , "r")  
    texte2 = f2.read()  
    f2.close()  
    if texte1 == texte2:  
        return True  
    return False
```

Exercice 4 : Ecriture d'un programme simple (4 points)

On suppose que vous avez un fichier nommé « coureurs.txt » qui contient la liste des coureurs participant au marathon annuel organisé à Béjaia (voir figure suivante). On vous demande d'écrire un programme qui :

- Définit une fonction nommée « **lireResultat()** » qui lit les résultats de la course et renvoi un dictionnaire ayant pour clé le nom du coureur et pour valeur son score.
- Définit une fonction nommée « **afficherClassement()** » qui affiche le classement des coureurs.



```
*coureurs.txt (~) - gedit
*coureurs.txt x
Mohand : 12
Said : 14
Mahmoud : 15
Tarik : 16
Fateh : 18
Lhadi : 12
Louisa : 12
```

```
"""
ce programme permet de :
---> lire les résultats d'un marathon
      numéro du coureur / score depuis un fichiers
---> d'afficher le classement des coureurs
"""

def lireResultats():
    f = open("coureurs.txt", "r")
    res = f.read()
    coureurs = res.split("\n")
    while "" in coureurs:
        coureurs.remove("")
    scoreDe = dict()
    for e in coureurs:
        print(e)
        nom = e.split(":")[0]
        s = e.split(":")[1]
        scoreDe[nom]=int(s)
    return scoreDe

def afficherClassement(scoreDe):
    coureurs = list(scoreDe.keys())
    scores = list(scoreDe.values())
    scoresTriés = list(scoreDe.values())
    scoresTriés.sort()
    print("voici les résultats : ")
    rang = 1
    for s in scoresTriés:
        ind = scores.index(s)
        print(rang, " : ", coureurs[ind], ": ", s )
        scores[ind] = max(scoresTriés)
        rang = rang+1

afficherClassement(lireResultats())
```