

EXERCICE 1: QCM (10 points)

Q1 : Que va renvoyer la commande suivante :

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

- **<class 'list'>** ✓
- <class 'tuple'>
- <class 'dict'>
- <class 'set'>

Q2 : Que va renvoyer la commande suivante :

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

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

Q3 : Que va afficher le code suivant :

```
>>> a = [ "test", [ 1 , 2 , 3 ] , [ 1 , 2 ] ]
>>> len(a)
```

3

Q4 : Que renvoi l'expression
"jsk" in ("Jsk","banane","pomme") :

False

Q5 : Les séquences immuables sont...:

- **Chaîne** ✓
- **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").Que renvoi l'instruction `fruits[1:2]` :

('pomme')

Un tuple !

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

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

5

Q9 : Soit une liste initialisée comme suit :

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

Quelle est la commande python permettant de créer, à partir de la variable « *listeMots* », une nouvelle variable « *phrase* » contenant une chaîne composée des éléments de la variable « *listeMots* » séparés par le caractère « : ».**phrase = ":".join(listeMots)**

Q10 : Soit les commandes suivantes :

```
>>> pays = dict()
>>> pays["Maghreb"] = [ "Algérie", "Tunisie" ]
>>> pays["Europe"] = [ "France", "Belgique" ]
>>> pays["Europe"] = [ "Allemagne", "Turquie" ]
>>> print(len(pays))
```

Que va afficher la dernière commande ?

2

Q11 : Que va afficher le code suivant ?:

```
def afficher():
    print ( " Bonjour " )
    print ( " mes amis " )
```

mes amis

Q12 : Soit les commandes suivantes :

```
>>> Club = ['JSK', 'MOB', 'JSMB', 'MCA']
>>> print ( Club[-4] )
```

Que va afficher la dernière commande ?

JSK

Q13 : Que va afficher le code suivant ?:

```
def afficher(nom="STID") :
    print ( " Bonjour " +nom)
    return
    print ( " mon amis " )
afficher()
```

Bonjour STID

Q14 : Que va afficher le code suivant ?:

```
>>> x = 18
>>> y = '20'
>>> print( float( ( str(x) + y)*2 ) + 0.5 )
```

18201820.5

Q15 : Que va afficher le programme suivant :

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

Béjaia ????

Q16 : Que va afficher le programme suivant :

```
v = 3
def f ( x , y = 2):
    x = x - y
    return x+v
print ( f ( v ) )
```

4

Q17 : Que va afficher le programme suivant :

```
v = 5
def f ( x, y = 2):
    x = x - y
    return x+v
print ( f ( 5, 6 ) )
```

4

Q18 : Que va afficher le programme suivant :

```
v = 5
def f ( x, y = 2):
    x = x + y
    return x + str ( v )
print ( f ( " Equipe ", " MOB
```

Equipe MOB 5

Q19 : Que va afficher le programme suivant :

```
D = dict()
def f(x):
    for i in range(3):
        D[i] = i*2
f(10)
print( list ( D.values() ) )
```

[0, 2, 4]

Q20 : Que va afficher le programme suivant :

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

[1, 2, 1, 3]

Problème (10 points) : On souhaite gérer un répertoire téléphone. A cet effet, on suppose que nous avons déjà un fichier nommé « *repertoire.txt* » contenant quelques numéros de téléphones avec leurs propriétaires comme indiqué sur la figure si-contre :

```
nom : téléphone
Bouzidi:1201020305
LHadi:4455223366
```

A - Écrire une fonction "*menu()*" permettant d'afficher un menu comme illustré sur la figure suivante :

```
Gestion d'un répertoire téléphonique
=====
1 - Ajouter un nouveau numéro
2 - Supprimer un numéro
3 - Quitter
=====

Choisissez entre 1 et 3 :
```

```
def menu():
    print("Gestion d'un répertoire téléphonique\n"+50*"=")
    print("1 - Ajouter un nouveau numéro")
    print("2 - Supprimer un numéro")
    print("3 - Quitter")
    print(50*"=")

    rep = input("Choisissez entre 1 et 3 : ")
    while rep not in ["1", "2", "3"]:
        rep = input(" ---> Vous devez choisir entre 1, 2 et 3 : ")
    return int(rep)
```

/ 2

Barème total sur 2 points	
Éléments d'évaluation (capacités attendues)	Éléments de notation
Écrire le code en respectant la syntaxe du langage	Ôter 0.5 à 2 point selon la gravité de l'erreur <ul style="list-style-type: none"> manque « : » par exemple ôter 0.5 point erreur d'indentation flagrante rendant le code complètement erroné, ôter 2 points erreur d'indentation moins flagrante ôter 0.5 points
Afficher le menu en conformité avec ce qui est demandé	Donnez 0.5 point si le menu affiché est conforme à ce qui est demandé sinon donner 0 point. Je rappelle que ce qui est demandé est comme suit : <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>Gestion d'un répertoire téléphonique ===== 1 - Ajouter un nouveau numéro 2 - Supprimer un numéro 3 - Quitter ===== Choisissez entre 1 et 3 :</pre> </div>
Demander à l'utilisateur de donner un choix et vérifier si ce choix est conforme à ce qui est demandé. Tant que l'utilisateur donne un mauvais choix, le code doit redemander à l'utilisateur de donner une autre choix	Donnez 1 si le code est conforme à ce qui est demandé, 0 sinon
Renvoyer le choix de l'utilisateur	Donnez 0.5 point s'il y a renvoi (correct) du résultat

B - Écrire une fonction « *ajouterNumero(fichier)* » permettant d'ajouter un nouveau numéro de téléphone avec son propriétaire dans le fichier « *fichier* : un fichier physique » indiqué comme paramètre.
 Attention, dans cette fonction, vous devez lire le nom du propriétaire et son numéro de téléphone puis vous devez vérifier si ce numéro existe dans le fichier physique « *repertoire.txt* » et ce n'est qu'à ce moment là que vous ajouterez ce nouveau numéro dans le fichier en question
Ne pas oublier bien évidemment d'ouvrir le fichier avec le bon mode d'ouverture et de le fermer après usage !

```
# Voici le code de la fonction "ajouterNumero(fichier)"
def ajouterNumero(fichier):
    """Cette fonction ajoute un nouveau numéro de téléphone
    dans un fichier de répertoire téléphonique"""

    # lire le numéro de téléphone et son propriétaire

    num = ""
    while not verifierNumTel(num):
        num = input("Donnez un numéro de téléphone sur 10 chiffres!")
        nom = input("Donnez le nom du propriétaire de ce numéro")

    # Ajouter ce numéro dans le répertoire tout en vérifiant s'il
    # existe pas déjà auquel cas, il ne faut pas le rajouter

    f = open(fichier, "r")
    rep = f.read()
    f.close()

    lignes = rep.split("\n")
    while "" in lignes:
        lignes.remove("")
    lignes.pop(0)

    # Vérifier si le numéro n'exite pas déjà
    numExistant = False
    for l in lignes:
        if num in l:
            numExistant = True
    if numExistant:
        print("Le numéro existe déjà")
    else :
        f = open(fichier, "a")
        f.write("\n"+nom+": "+num)
        f.close()
```

3

Barème total sur 3 points		
Éléments d'évaluation	barème	Éléments de notation
Écrire le code en respectant la syntaxe du langage	-2 à 0 points	Ôter 0.5 à 2 points selon la gravité de l'erreur <ul style="list-style-type: none"> manque « : » par exemple ôter 0.5 point erreur d'indentation flagrante rendant le code complètement erroné, ôter 2 points erreur d'indentation moins flagrante ôter 0.5 points
Lecture avec vérification du numéro de téléphone	0 à 1 point	Donnez 1 point si c'est conforme à ce qui est demandé Donnez 0.5 point si pas de vérification des données lues au clavier (chaîne composée de 10 chiffres)
Ouvrir un fichier en lecture et récupérer son contenu brut dans une variable « string »	0 à 0.5 point	La fonction open doit être utilisée correctement (bonne indication du fichier physique et du mode d'ouverture) Fermeture du fichier (méthode close()) utilisation de la méthode « read() »
Convertir un texte en une liste et purifier cette liste en supprimant les informations inutiles et problématiques	0 à 0.5 points	L'étudiant doit convertir le texte en une liste pour récupérer les lignes du fichier. Il doit aussi supprimer la première ligne et les lignes vides de cette liste. Donnez 0.5 points si ces 2 conditions ont été atteintes, 0 sinon
Vérifier si le numéro à ajouter existe auquel cas il faut le supprimer. Si le numéro n'existe pas il faut l'ajouter au fichier, donc être capable de l'ouvrir avec le bon mode d'ouverture et lui ajouter le nouveau numéro	0 à 1 point	Donnez 0.5 si la vérification de l'existence du numéro dans le répertoire téléphonique est correcte Donnez 0.5 si l'ajout du numéro dans le répertoire téléphonique est correct. Si les conditions ci-dessus ne sont pas satisfaites donnez 0 point

C : Écrire une fonction « *supprimerNumero(fichier)* » permettant de supprimer un numéro de téléphone d'un répertoire téléphone (*fichier* : un fichier physique) donné en argument à la fonction.

Attention, dans cette fonction, vous devez :

- lire un numéro de téléphone au clavier
- ouvrir le fichier physique avec le bon mode d'ouverture
- vérifier si le numéro de téléphone existe ou non dans le fichier physique
- supprimer la ligne contenant le numéro de téléphone à supprimer
- mettre à jours le fichier physique de sorte à ce qu'il prenne en compte la suppression du numéro

Ne pas oublier bien évidemment d'ouvrir le fichier avec les bons modes d'ouverture et de le fermer après chaque usage !

```
def supprimerNumero(fichier):

    # lire le numéro de téléphone à partir du clavier. Ne pas oublier de vérifier la saisie de
    # Une chaîne de caractères composée de 10 chiffres! pour cela je fait appel à la fonction v
    num = ""
    while not verifierNumTel(num): num = input("Donnez un numéro de téléphone sur 10 chiffres!")

    f = open(fichier, "r") # Ouvrir le fichier physique en lecture
    rep = f.read() # Lire tous les numéros de téléphones du répertoire dans la variable "rep"
    f.close() # ne pas oublier de fermer le fichier

    lignes = rep.split("\n") # extraire les lignes de ce répertoire dans la variables "lignes"
    # chaque ligne correspond respectivement à un nom et un numéro de téléphone

    ligneUne = lignes[0] # Récupérer provisoirement la première ligne qui correspond à l'entête du fichier
    lignes.pop(0) # Récupérer provisoirement la première ligne qui correspond à l'entête du fichier

    while "" in lignes: # supprimer toutes les lignes vides
        lignes.remove("")

    # Vérifier si le numéro n'exite pas déjà
    indice = 0 ; numExistant = False
    while True :
        if num in lignes[indice]:numExistant = True; lignes.pop(indice); break
        indice = indice+1

    if numExistant:
        lignes.insert(0,ligneUne) # Reconstruire le contenu du fichier sans le numéro dans
        contenuFichier = "\n".join(lignes) # une variable nommée "contenuFichier"
        f = open(fichier, "w") # Ouvrir le fichier physique avec l'option d'accès "w".
        f.write(contenuFichier) # Ecrire le nouveau contenu dans le fichier physique
        f.close() # ne pas oublier de fermer le fichier
    else : print("votre numéro est inexistant dans le répertoire !") # Si le numéro n'existe pas dans le
    # répertoire téléphone il faut juste avertir l'utilisateur que le numéro donné est inexistant
    # dans le répertoire téléphonique
```

Barème total sur 3 points

Éléments d'évaluation	barème	Éléments de notation
Écrire le code en respectant la syntaxe du langage	-2 à 0 points	Ôter 0.5 à 2 points selon la gravité de l'erreur <ul style="list-style-type: none"> • manque « : » par exemple ôter 0.5 point • erreur d'indentation flagrante rendant le code complètement erroné, ôter 2 points • erreur d'indentation moins flagrante ôter 0.5 points
Lecture avec vérification du numéro de téléphone	0 à 1 point	Donnez 1 point si c'est conforme à ce qui est demandé Donnez 0.5 point si pas de vérification des données lues au clavier (chaîne composée de 10 chiffres)
Ouvrir un fichier en lecture et récupérer son contenu brute dans une variable « string »	0 à 0.5 point	La fonction open doit être utilisée correctement (bonne indication du fichier physique et du mode d'ouverture) Fermeture du fichier (méthode close()) utilisation de la méthode « read() »
Convertir un texte en une liste et purifier cette liste en supprimant les informations inutiles et problématiques	0 à 0.5 points	L'étudiant doit convertir le texte en une liste pour récupérer les lignes du fichier. Il doit aussi supprimer la première ligne et les lignes vides de cette liste. Donnez 0.5 points si ces 2 conditions ont été atteintes, 0 sinon
Vérifier si le numéro à ajouter existe déjà auquel cas il faut le supprimer du fichier (en ouvrant avec le bon mode le fichier physique). Si le numéro n'existe pas il faut avertir de cela l'utilisateur	0 à 1 point	Donnez 0.5 si la vérification de l'existence du numéro dans le répertoire téléphonique est correctement Donnez 0.5 si la suppression du numéro du répertoire téléphonique est correct. Si les conditions ci-dessus ne sont pas satisfaites donne 0 point

D : Écrire un programme principal permettant :

- d'afficher le menu réalisé par la fonction « *menu()* »
- d'ajouter, de supprimer un numéro de téléphone ou de quitter le programme selon le choix de l'utilisateur

```
# Programme principal
# =====

# indique le fichier physique du répertoire téléphone
fichier = "repertoire.txt"

# Boucler tanque l'utilisateur ne demande pas
# explicitement de quitter en choisissant le menu 3
while True:

    # Récupérer le choix de l'utilisateur
    m = menu()

    # Effectuer le traitement adéquat selon le choix de l'utilisateur
    if m==1:
        print(40*" "+"1 - Ajouter un nouveau numéro\n")
        ajouterNumero(fichier)
        input("tapez un touche pour continuer ....")
    if m==2:
        print(40*" "+"2 - Supprimer un numéro")
        supprimerNumero(fichier)
        input("tapez un touche pour continuer ....")
    if m==3:
        print("Fin du programme")
        input("tapez un touche pour quitter ....")
        break
```

Barème total sur 2 points		
Éléments d'évaluation	barème	Éléments de notation
Écrire le code en respectant la syntaxe du langage	-2 à 0 points	Ôter 0.5 à 2 points selon la gravité de l'erreur <ul style="list-style-type: none"> • manque « : » par exemple ôter 0.5 point • erreur d'indentation flagrante rendant le code complètement erroné, ôter 2 points • erreur d'indentation moins flagrante ôter 0.5 points
Indiquer le nom du fichier physique dans un variable	0 à 0.5 points	
Boucler jusqu'à ce que l'utilisateur demande de quitter afficher un menu et selon le choix de l'utilisateur effectuer un traitement adéquat.	0 à 1.5 points	