

EMD Langage Evolué 1 | Nom : Prénom : Groupe :

EXERCICE 1 : QRC et QCM (12 points)

Q1 : Parmi les méthodes ci-dessous, indiquez celle(s) qui supprime(nt) le dernier élément d'une liste :

index() insert() pop() remove()

Q2 : Parmi les identificateurs ci-dessous, indiquez ceux qui correspondent à des méthodes de la classe « dict() » :

del() sort() items()
 index() max() get()
 insert() count() len()
 pop() keys() sqrt()
 remove() values() import

Q3 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Algérie"
>>> s.startswith("al")
```

Q4 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Algérie 2016"
>>> s.isdigit()
```

Q5 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Algérie 2016"
>>> print(s[1:3]*2)
```

Q6 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Bejaia 2018"
>>> print(s.count(s[3]))
```

Q7 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Bejaia 2018"
>>> v = s.replace("a", "b")
>>> print(s)
```

Q8 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Bejaia 2018"
>>> v = s.replace("a", "b")
>>> print(v)
```

Q9 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Bejaia 2018"
>>> print(s[-4:-1])
```

Q10 : Indiquez ce que va afficher le code suivant:

```
>>> s = "Bejaia 2018"
>>> print(s[-4:][0:3])
```

Q11 : Indiquez ce que va afficher le code suivant:

```
>>> serie = list(range(1, 10, 2))
>>> print(serie)
```

Q12 : Indiquez ce que va afficher le code suivant:

```
>>> serie = list(range(10, 1, -3))
>>> print(serie[1:])
```

Q13 : Indiquez ce que va afficher le code suivant:

```
>>> serie = list(range(20, 5, -5))
>>> print(serie[-3:])
```

Q14 : Donnez la commande permettant d'insérer l'objet « Banane » juste après l'objet « Melon » :

```
>>> fruits=["Orange", "Melon", "Pastèque"]
```

Q15 : indiquez ce que va afficher le code suivant:

```

1 while True:
2     x = 0
3     liste = []
4     while True:
5         if x == 5:
6             break
7         liste.append(x)
8         x = x+1
9     print(liste)
10    if x ==5:
11        break

```

.....

Q16 : Indiquez ce que va afficher le code suivant :

```

1 monaies = ["Dinare", "Dollar", "Euro"]
2 m = []; i = 0
3 while True:
4     if i == len(monaies):
5         break
6     if monaies[i].startswith("D"):
7         m.append(monaies[i])
8     i=i+1
9 print(m)

```

.....

Q17 : Donnez la commande permettant d'importer toutes les fonctions du module « `math` » :

.....

Q18 : Donnez la commande permettant d'importer le module « `math` » sous l'aléa « `mt` » :

.....

Q19 : Indiquez ce que va afficher le code suivant :

```

1 eleves = ["E1", "E2", "E3"]
2 notesDe = {}
3 notes = ([14,15], [13], [12])
4 i = 0
5 for e in eleves:
6     notesDe[e] = notes[i]
7     i = i+1
8 print(notesDe)

```

.....

Q20 : Indiquez ce que va afficher le code suivant :

```

1 A = ["X", "T", "A", "C", "K"]
2 B = {}
3 for i in range(3):
4     B[i] = A[i]
5 print(B)

```

.....

Q21 : Indiquez ce que va afficher le code suivant :

```

1 A = [14]
2 def f(x,y=[1]):
3     x = x*2+A+y
4     return x
5 print(f(A))

```

.....

Q22 : Indiquez ce que va afficher le code suivant :

```

1 A = max([14,16])
2 def f(x,y=[1]):
3     x = x*2+A+y
4     return x
5 print(f(4,1))

```

.....

Q23 : Indiquez ce que va afficher le code suivant :

```

1 def f(x,y=[1]):
2     x = x*2+y
3     return x
4 print(f("4","1"))

```

.....

Q24 : Indiquez ce que va afficher le code suivant :

```

def f(x, y):
    if type(x) == type(y) :
        if type(x) == str:
            return (x + "=" + y)
        elif type(x) == int or type(x) == float:
            return x*y
    else:
        return "rien"
print(f(12,"10"))

```

.....

Question Bonus : indiquez ce que va afficher le code suivant:

```

def f(x, y):
    if type(x) == type(y) :
        if type(x) == str:
            return (x + "=" + y)
        elif type(x) == int or type(x) == float:
            return max([x,y])
    else:
        return "rien"
print(f(12,3))

```

.....

PROBLEME (8 points)

On suppose que vous avez un fichier nommé « **factures.txt** » se trouvant dans le même dossier que votre programme et contenant un texte représentant des informations sur la facture d'un client. La première ligne contient le nom du client précédé d'un « # ». Les autres lignes contiennent chacune : le nom, la quantité et le prix unitaire du produit commandé. Voir illustration ci-contre :

```
# Client 1
Imprimante : 12 : 14000 DA
Onduleurs : 15 : 10000 DA
Ordinateur portable : 2 : 50000 DA
```

On vous demande d'écrire un programme permettant ceci :

- ❶ Afficher le menu suivant

```
Gestion des factures
=====
1 - afficher le nom du client
2 - afficher les produits commandés
3 - afficher le montant de la facture
4 - Quitter

Donnez votre choix (1 à 4)?
|
```

Indication : Utilisez une fonction nommée « **menu()** »

- ❷ Afficher le nom du client, si l'utilisateur choisi le menu 1:

```
Gestion des factures
=====
1 - afficher le nom du client
2 - afficher les produits commandés
3 - afficher le montant de la facture
4 - Quitter

Donnez votre choix (1 à 4)? 1
=====

Le nom du client est Client 1

Tapez une touche pour continuer...
|
```

Indication :

- Utilisez une fonction nommée « **nomClient(text)** ».
- Le paramètre « **text** » correspond au texte brute du fichier « **facture.txt** »

- ❸ Afficher la liste des produits commandés, si l'utilisateur choisi le menu 2 :

```
Gestion des factures
=====
1 - afficher le nom du client
2 - afficher les produits commandés
3 - afficher le montant de la facture
4 - Quitter

Donnez votre choix (1 à 4)? 2
=====

Voici la liste des produits commandés:
['Imprimante', 'Onduleurs', 'Ordinateur portable']

Tapez une touche pour continuer...
|
```

Indication :

- Utilisez une fonction nommée « **listeProduits(lignesFacture)** ».
- Le paramètre « **lignesFacture** » correspond à une liste contenant toutes les lignes du texte brute du fichier « **facture.txt** » à l'exception de la première

- ❹ Calculer le montant total de la facture (somme des quantités multipliées par les prix unitaires), si l'utilisateur choisi le menu 3 :

```
Gestion des factures
=====
1 - afficher le nom du client
2 - afficher les produits commandés
3 - afficher le montant de la facture
4 - Quitter

Donnez votre choix (1 à 4)? 3
=====

Le montant de la facture est : 418000.0 DA

Tapez une touche pour continuer...
|
```

- ❺ S'arrête si l'utilisateur choisi le menu 4.

Nom : Prénom :Groupe :

Définition de la fonction **menu()** *#cette fonction ne reçoit rien, mais retourne un entier*

/ 1

Définition de la fonction **nomClient(...)** *# cette fonction reçoit un texte (un string correspondant au texte brut du fichier # « facture.txt ») et retourne un nom de client (un string)*

/ 1

Définition de la fonction **listeProduits(...)** *# cette fonction reçoit une liste de chaines contenant chacune un nom de produits, sa quantité et son prix unitaire et en retour elle renvoi uniquement # les noms des produits*

/ 2

Définition de la fonction `montantFact(.....)` *# cette fonction reçoit une liste de chaînes contenant chacune un nom de produits, sa quantité et son prix unitaire et en retour elle renvoie le montant de la facture qui correspond à la somme des quantités multipliées par les prix unitaires*

/ 2

Programme principal ()

```
f = open("facture.txt", "r")
text = f.read()
```

```
# extraction des lignes de la facture
lignesFacture = text.split("\n")
lignesFacture.pop(0)
while "" in lignesFacture:
    lignesFacture.remove("")
```

/ 2

La variable « text » contient le texte brut du fichier « facture.txt »

La variable « lignesFacture » contient les lignes de la facture à l'exception de la première