

Examen :

Exercice 03 :

On se propose d'écrire une application en python qui permet de gérer les factures d'un magasin qui n'ont pas encore été payées. Chaque facture est définie par une liste comme dans l'exemple suivant :

```
facture=[1, 'Ahmed Naciri', 354.00, 2019]
```

dont *facture[0]* est le numéro de la facture, *facture[1]* le nom du client, *facture[2]* le prix hors taxe (prix HT) de la facture et *facture[3]* l'année limite de paiement.

1. Définir la fonction *lire()* permettant de créer et de retourner une facture (liste) dont le *numéro*, *nom*, *prix HT* et *année limite* sont lus au clavier
2. En utilisant la fonction *lire()*, définir la fonction *lire_tous(n)* permettant de lire les *n* factures non payées, dont leurs informations sont lues au clavier, de les stocker dans une liste et de la retourner.
3. Définir la fonction *trier(factures)* permettant de trier la liste des factures par ordre croissant des noms des clients.

4. En supposant que la liste des factures est triée par ordre croissant des noms des clients, et en utilisant une recherche dichotomique, définir la fonction *recherche_dichotomique(factures, nom_client)* retournant la facture correspondant au client dont le nom est passé en argument et liste vide sinon.

5. Ecrire une fonction d'entête *maj_factures(factures)* qui permet d'augmenter de 10% par an de retard les prix HT de toutes les factures dont l'année limite de paiement est strictement inférieure à l'année en cours (2020)

Exemple : Soit la facture ayant l'année limite de paiement 2018 et le prixHT=100dh, le nouveau prix deviendra 121 dh. (De 2018 à 2019 on augmente le prix de 10% soit 110, de 2019 à 2020, on augmente 110dh de 10% soit 121 dh)

6. Ecrire la fonction *total_TTC(factures)* qui retourne la somme des prix totaux TTC (Toute Taxe Comprise) de toutes les factures dans la liste *factures*. La TVA étant fixée à 20% (0.2) ($Prix_TTC = Prix_HT + Prix_HT * TVA$)
7. Définir la fonction *sauvgarder(factures, nom_fichier)* permettant de sauvegarder toutes les factures dans le fichier texte dont le nom est passé en argument.
8. Ecrire une fonction principale de test, où vous appelez toutes les fonctions définies plus haut, avec des exemples d'utilisation de votre choix.

```
def lire():
    num=int(input("Entrer le numéro du client: "))
    nom=input("Entrer le nom du client: ")
    pHT=float(input("Entrer le prix HT: "))
    annee=int(input("Entrer l'annee limite: "))
    L=[num,nom,pHT,annee]
    return L

def lire_tous(n):
    L=[]
    for i in range(n):
        L.append(lire())
    return L

#methode1 de tri
def trier(L):
    L.sort(key=lambda x:x[1])
    return L

#methode2 de tri
def trier2(L):
    for j in range(len(L)-1,0,-1):
        for i in range(0,j,1):
            if L[i][1]>L[i+1][1]:
                k=L[i]
                L[i]=L[i+1]
                L[i+1]=k
    return L
```

```
def rechDicho(L,nom):
    m=len(L)//2
    if L[m][1]==nom:
        return L[m]
    else:
        if nom<L[m][1]:
            return rechDicho(L[:m],nom)
        else:
            return rechDicho(L[m+1:],nom)

#rechercheSequentielle
def rechSeq(L,nom):
    for x in L:
        if x[1]==nom:
            return x
    return "Client n'existe pas"

def maj_factures(L):
    n=len(L)
    for i in range(n):
        if L[i][3]<2020:
            c=2020-L[i][3]
            for k in range(c):
                L[i][2]=(L[i][2]*10)/100+L[i][2]
```

```
def total_ttc(L):
    s=0
    for i in range(len(L)):
        s+=L[i][2]*0.2+L[i][2]
    return s
```

```
def sauvegarder(L,fich):
    f=open(fich,"w")
    f.write(str(L))
    f.close()
```

```
>>> L=lire_tous(3)
Entrer le numéro du client: 5
Entrer le nom du client: bachar
Entrer le prix HT: 150
Entrer l'annee limite: 2018
Entrer le numéro du client: 7
Entrer le nom du client: alami
Entrer le prix HT: 140
Entrer l'annee limite: 2021
Entrer le numéro du client: 1
Entrer le nom du client: halim
Entrer le prix HT: 170
Entrer l'annee limite: 2022
>>> trier2(L)
[[7, 'alami', 140.0, 2021], [5, 'bachar', 150.0, 2018], [1, 'halim', 170.0, 2022]]
>>> rechDicho(L,'alami')
[7, 'alami', 140.0, 2021]
>>> maj_factures(L)
>>> L
[[7, 'alami', 140.0, 2021], [5, 'bachar', 181.5, 2018], [1, 'halim', 170.0, 2022]]
>>> total_ttc(L)
589.8
>>> sauvegarder(L,'facture.txt')
```

