Année universitaire : 2024 / 2025 **Pr. Cherrat Loubna**

Algorithmes et programmation Python

Semestre S6

Série de TD N° : 3

Chaines de caractères & listes

Exercice 1 :

Écrire un script python qui demande à l'utilisateur de saisir une chaine sous forme d'une phrase et retourne le nombre de mots constituant cette phrase. Si la chaîne est vide, il affiche un message d'erreur.

```
# Fonction qui retourne le nombre de mots dans une chaîne

def compter_mots(chaine):

    # Vérifier si la chaîne n'est pas vide

    if chaine.strip():  # Utilisation de strip() pour ignorer les espaces autour de la chaîne

        mots = chaine.split()  # Diviser la chaîne en mots

        return len(mots)  # Retourner le nombre de mots

    else:

        return 0  # Retourner 0 si la chaîne est vide

# Demander à l'utilisateur de saisir une chaîne

phrase = input("Veuillez saisir une phrase : ")

# Appeler la fonction et afficher le résultat

nombre_de_mots = compter_mots(phrase)

if nombre_de_mots == 0:

    print("La chaîne saisie est vide ou contient uniquement des espaces.")

else:
```

```
    print(f"Le nombre de mots dans la phrase est : {nombre_de_mots}")
```

Exercice 2 :

Écrire une fonction Python qui prend en paramètre une chaîne, la transforme en majuscules, puis trie les caractères par ordre alphabétique et retourne la nouvelle chaîne triée

```
# Fonction qui prend une chaîne, la met en majuscules et la trie par ordre alphabétique

def chaine_triee(chaine):

    # Mettre la chaîne en majuscules

    chaine_maj = chaine.upper()

    # Trier les caractères par ordre alphabétique

    chaine_triee = ''.join(sorted(chaine_maj))

    #chaine_triee = sorted(chaine_maj)

    return chaine_triee

# Exemple d'utilisation

chaine = "salut les programmeurs"

resultat = chaine_triee(chaine)

print(chaine , ",après trie est :",resultat)

# Affiche " AEGLLMNOOPRRSSSTU"
```

Exercice 3 :

Écrire une fonction **Élimination** qui prend en entrée une liste L et un entier i, et qui retourne la liste dans laquelle on a éliminé la case d'indice i.

```
def Elimination(L, i):
    """Supprime l'élément d'indice i de la liste L et retourne la nouvelle liste"""
    if 0<= i < len(L):  # Vérifie si l'indice est valide
     L.pop(i) #  OU del L[i] qui Supprime aussi l'élément à l'indice i
    else:
        print("Indice invalide !")
    return L
```

```
# Exemple d'appel 1
ma_liste = [10, 20, 30, 40, 50]
# Saisie de l'indice à supprimer
i = int(input("Entrez l'indice de l'élément à supprimer : "))

nouvelle_liste = Elimination(ma_liste, i)
print("Liste après suppression :", nouvelle_liste)
# Exemple d'appel 2

# Saisie de la liste par l'utilisateur
ma_liste = []
# Demander à l'utilisateur combien de nombres il veut ajouter

n = int(input("Combien de nombres voulez-vous ajouter à la liste ? "))

# Demander à l'utilisateur de saisir chaque nombre

for i in range(n):
    valeur = int(input(f"Saisissez le {i+1}-ème nombre: "))
    ma_liste.append(valeur)
i = int(input("Entrez l'indice de l'élément à supprimer : "))

# Appel de la fonction et affichage du résultat
nouvelle_liste = Elimination(ma_liste , i)
print("Liste après suppression :", nouvelle_liste)
```

V2

```
ef Elimination(L, i):
    """Supprime l'élément d'indice i de la liste L et retourne la nouvelle liste"""
    if 0 <= i < len(L):  # Vérifie si l'indice est valide
        del L[i]  # Supprime l'élément à l'indice i
    else:
        print("Indice invalide ! Aucun élément supprimé.")
    return L

# Saisie de la liste par l'utilisateur
L_str = input("Entrez les éléments de la liste séparés par des espaces : ").split()
L = []  # Liste vide

for valeur in L_str:
    L.append(int(valeur))  # Convertir chaque valeur en entier et l'ajouter à la liste

# Saisie de l'indice à supprimer
```

```
i = int(input("Entrez l'indice de l'élément à supprimer : "))

# Appel de la fonction et affichage du résultat
nouvelle_liste = Elimination(L, i)
print("Liste après suppression :", nouvelle_liste)
```

Exercice 4 :

Fonction demandée :

```
def RechercheTousIndices(x, L):
    indices = tuple(i for i, val in enumerate(L) if val == x)
    return indices
```

OU ----------V2_____\\\888

```
def RechercheTousIndices(x, L):
    indices = []
    for i in range(len(L)):
        if L[i] == x:
            indices.append(i)
    return tuple(indices)
#APPEL
liste = [4, 7, 4, 3, 4, 9]
valeur = 4
resultat = RechercheTousIndices(valeur, liste)
print("Les indices de", valeur, "sont :", resultat)
```

\\\\\VERSION AVEC SAISIE

Version interactive (avec saisie utilisateur) :

```
def RechercheTousIndices(x, L):
    indices = tuple(i for i, val in enumerate(L) if val == x)
    return indices
```

```
# Saisie utilisateur
chaine = input("Entrez une liste d'entiers séparés par des virgules : ")
liste = [int(val) for val in chaine.split(',')]
x = int(input("Entrez la valeur à rechercher : "))

# Appel de la fonction
indices_trouves = RechercheTousIndices(x, liste)

# Affichage du résultat
if indices_trouves:
    print("□ Indices de", x, ":", indices_trouves)
else:
    print("□ Valeur non trouvée dans la liste.")
```

Exercice 5 :

On considère un dictionnaire **Notes** contenant les notes obtenues par un groupe d'étudiants dans un module donnée.

Les clés de ce dictionnaire sont représentées par les noms des étudiants tandis que les valeurs des clés sont représentées par les moyennes générales obtenues.

Écrire un script python qui permet de :

1. Remplir ce dictionnaire par les **noms** ainsi que les **notes** de **N** étudiants (N est choisi par l'utilisateur) .
2. Afficher la liste des étudiants avec leurs notes triée par ordre alphabétique.
3. Diviser ce dictionnaire en deux sous dictionnaires (**valides, non_valides**) : le premier contient les étudiants ayant validé ce module, le deuxième contient les étudiants n'ayant pas validés ce module.Puis, Afficher les deux dictionnaires.
4. Rechercher puis afficher le nom de l'étudiant ayant la note maximale.

➢ **SOLUTION**

```
# Question 1
Notes={}
N=int(input("Nombre des étudiants : "))
for i in range (N) :
  print("Etudiant " , i+1)
  nom=input("Nom = ")
  note=float(input("Note = "))
  Notes[nom]=note
# Question 2
```

```
print("Liste des étudiants par ordre alphabétique : ")
for x in sorted(Notes.keys()):
 print(x, Notes[x])
# Question 3
V={}
NV={}
for x,y in Notes.items():
    if y>=10:
       V[x]=y
    else :
      NV[x]=y
print("Les étudiants qui ont validé le module : ", V)
print("Les étudiants qui n'ont pas validé le module : ", NV)

# Question 4
def max_notes(dict_notes):
 L=[]
 for a,b in dict_notes.items() :
  if b==max(dict_notes.values()) : L.append(a)
 return L
print("Les étudiants ayant la note maximale sont : ")
print(max_notes(Notes))
```

Exercice 6:

Écrire un script python qui permet de saisir **N** valeurs numériques, puis les classer dans un dictionnaire contenant **deux** listes : une liste des **valeurs positives** et une deuxième liste des **valeurs négatives**.

Dictionnaire={"valeurs positives": [], "valeurs négatives": []}

- Le script affiche un message approprié dans le cas d'une valeur nulle
- **SOLUTION**

```
Nombres={ "Positif" : [],"Négatif" : []}

N=int(input("Entrer le nombre des valeurs à classer : "))

for i in range(N) :

    n=int(input ("entrer un nombre : "))

    if n>0 :

        Nombres["Positif"].append(n)
```

```
    elif n<0 :
      Nombres["Négatif"].append(n)
    else :
      print("Vous avez tapé une valeur nulle ")
print(Nombres)
```