

**TP 4 – LES LISTES****Préambule :**

1. Ouvrir une session sur l'ordinateur en entrant Identifiant / Mot de passe ;
2. Lancer Spyder en cherchant Spyder dans la barre de recherche ;
3. Effacer le texte dans la fenêtre de gauche et écrire :

```
#NOM
#PRENOM
#TP4
```

**Objectifs de ce TP :**

- Manipuler des listes ;
- Calculer sur les listes d'entiers et de flottants ;
- Ecriture de fonctions associées ;

**Toutes les réponses aux questions doivent être écrites dans le programme qui sera à rendre dans le dossier Restitution de devoirs de votre classe avant 18h.**

**Exercice 1 : travail sur les listes**

1. Tester les commandes suivantes dans le terminal :

```
>>>var1=range(0,12)
```

```
>>>var2=list(var1)
```

```
>>>type(var2)
```

```
>>>print(var2[0])
```

```
>>>print(var2[5])
```

```
>>>print(var2[12])
```

```
>>>len(Var2)
```

Puis :

```
>>>var2=list(3,'bonjour',45.56)
```

```
>>>type(var2)
```

```
>>>len(Var2)
```

```
>>>print(var2[0])
```

```
>>>print(var2[1])
```

**Exercice 2 : accès aux éléments d'une liste**

L'accès aux éléments d'une liste se fait par :

```
L[début : fin+1 : pas]
```

En lisant la liste de gauche à droite pour un pas positif, ou de gauche à droite pour un pas négatif :

- « début » pour le premier élément retenu,
- « fin+1 » pour le premier élément **non retenu**,
- « pas » entre 2 valeurs retenues.

1. Ecrire à la suite du code le commentaire #Exercice 2.
2. Taper ces lignes et indiquer les résultats des `print` en commentaires sur chaque ligne :

```
liste = [00 ,10 ,20 ,30 ,40 ,50 ,60]
print(liste,type(liste))
print (liste[:4:1])
print (liste[4:10:2])
print (liste[5:3:-1])
```

3. Taper ces lignes et indiquer les résultats des `print` en commentaires sur chaque ligne :

```
mot = "vivement_noel"
print (type(mot))
list1 = list(mot)
print(list1,type(list1))
print (list1[:4:1])
print (list1[4:10:3])
print (list1[10:6:-1])
```

### Exercice 3 : commandes sur les listes

1. Ecrire à la suite du code le commentaire #Exercice 3.
2. Créer la liste suivante :

```
liste1 = [10 ,20 ,30 ,40 ,50 ,60 ,70 ,80 ,90 ,100]
```

3. Exécuter chaque bloc d'instructions et indiquer ce qui est fait sur la liste en commentaires :

Bloc 1 :

```
liste1 . append (33)
print ( liste1 )
```

Bloc 2 :

```
liste1 .pop (10)
print ( liste1 )
```

Bloc 3 :

```
liste1 . remove (40)
print ( liste1 )
```

Bloc 4 :

```
k = 3
del( liste1 [k])
print ( liste1 )
```

Bloc 5 :

```
liste1 . insert (3 ,40)
print ( liste1 )
```

Bloc 6 :

```
print (len( liste1 ))
```

Bloc 7 :

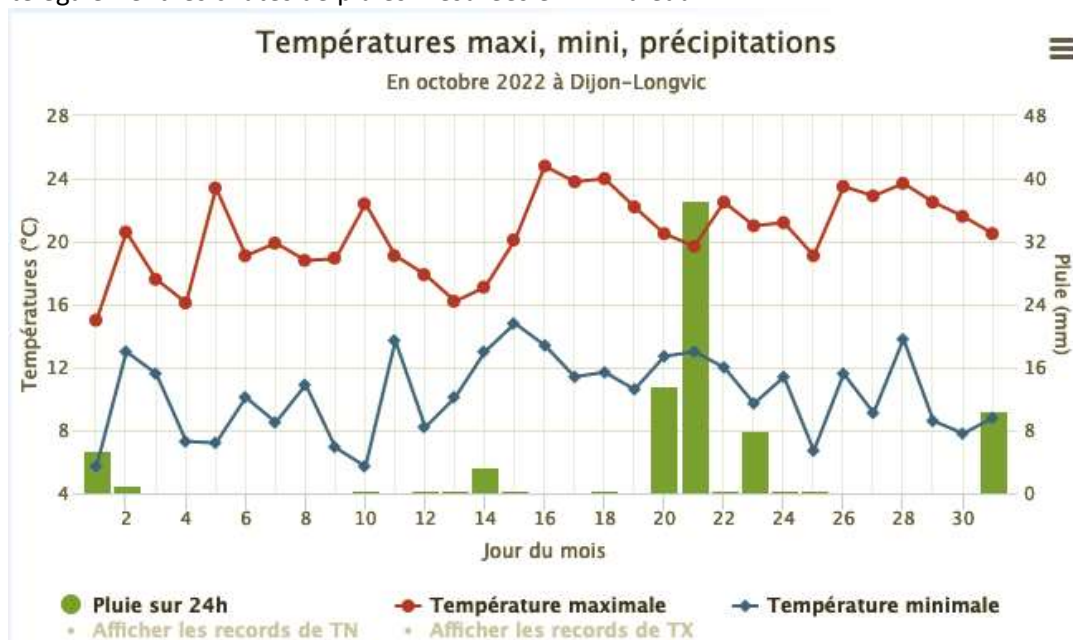
```
liste2 =[10 ,5 ,69 ,450 ,1 ,2]
liste2 .sort ()
print ( liste2 )
```

Bloc 8 :

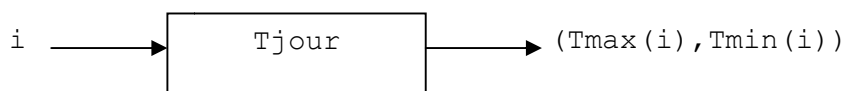
```
liste2 = liste2[::-1]
print (liste2)
```

**Exercice 4 : le changement climatique**

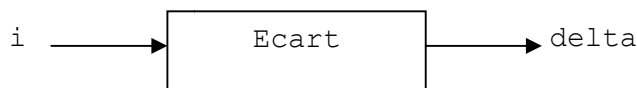
Le graphe ci-contre est le relevé des températures maximales et minimales quotidiennes pour le mois d'Octobre 2022 à Dijon. Il présente également les chutes de pluies mesurées en mm d'eau.



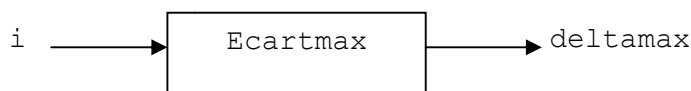
1. Ecrire à la suite du code le commentaire `#Exercice 4`
2. En utilisant le graphe, créer 2 listes appelés `Tmax` et `Tmin` contenant respectivement les températures quotidiennes maximales et minimales.
3. Vérifier avec une fonction que vos 2 listes contiennent bien 31 éléments chacune.
4. Créer une fonction `Tjour`, prenant pour argument le numéro `i` du jour dans le mois et renvoie les températures maximale et minimale du jour `i` :



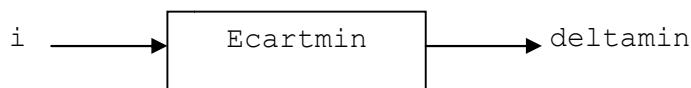
5. Créer une fonction `Ecart`, prenant pour argument le numéro `i` du jour dans le mois et renvoie `delta`, l'écart de température entre la maximale et la minimale du jour `i` :



5. Créer une fonction `Ecartmax`, prenant pour argument le numéro `i` du jour dans le mois et renvoie `deltamax`, l'écart de température entre la maximale du jour `i` et la maximale du mois d'Octobre :

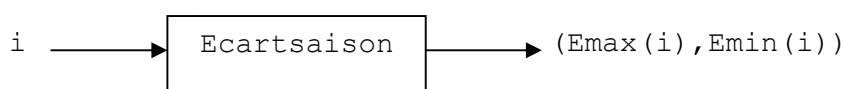


6. Créer une fonction `Ecartmin`, prenant pour argument le numéro `i` du jour dans le mois et renvoie `deltamin`, l'écart de température entre la minimale du jour `i` et la minimale du mois d'Octobre :



**La température de normale saisonnière maximale de 15.7 °C et minimale de 7°C pour un mois d'Octobre « normal ».**

7. Créer une fonction `Ecartsaison`, prenant pour argument le numéro `i` du jour dans le mois et renvoie `deltasaison`, les écarts de température maximale et minimale entre le jour `i` et la moyenne de saison :



8. Ecrire un programme qui demande à l'utilisateur (fonction `input`) le numéro `i` du jour à étudier et qui indique distinctement le résultat suivant :

Le **xx** Octobre 2022 :

la température maximale fut de : **xxxxx** °C ; avec un écart de **xxxx** °C avec la maximale mensuelle et de **xxxx** °C avec la normale saisonnière ;

la température minimale fut de : **xxxxxx** °C ; avec un écart de **xxxx** °C avec la minimale mensuelle et de **xxxx** °C avec la normale saisonnière.

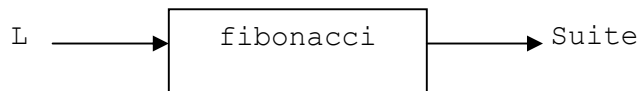
9. Tester et vérifier votre programme pour `i=3`, `11` et `26`.

### Exercice 5 : la suite de Fibonacci

La **suite de Fibonacci** est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence généralement par les termes 0 et 1, et ses premiers termes sont alors : 0, 1, 1, 2, 3, 5, 8, 13, 21, etc... Mais on peut choisir n'importe quels couples d'entiers pour définir les 2 premiers termes de la suite.



1. Ecrire à la suite du code le commentaire `#Exercice 5`.
2. Donner les 10 premiers termes de la suite de Fibonacci qui commence par les termes 1 et 1.
3. Créer une liste `L` qui contient les 2 premiers termes donnés par l'utilisateur. (utiliser pour cela la fonction `input`).
4. Tester votre code en ajoutant la ligne de code à la suite de votre programme : `print(L)`.
5. Créer la fonction `fibonacci`, qui prendra comme argument la liste `L` des 2 premiers termes, et renverra une liste `Suite` contenant les 10 termes suivants de la suite de Fibonacci.



6. Tester votre fonction avec la question 2.