-------------------------------------------------------------------------------------------------------------------------

***Exercice* : Ecrire une fonction *nb0(n)* qui retourne le nombre de zéros dans un entier. (sans utiliser str()).**

***Problème :*** ---------------------------------------------------------------------------------------

La température minimale (abréviation TN) est la plus basse observée entre 18 heures U.T.C. la veille et 18 heures le jour même.

La température maximale (abréviation TX) est la plus élevée observée entre 6 heures U.T.C. et le lendemain 6 heures. Elle se produit le plus souvent en cours d'après-midi.

La différence entre le minimum et le maximum d'une journée est appelée amplitude diurne. La température minimale est généralement plus élevée lorsque le temps est couvert, et la maximale au contraire plus élevée par temps clair. Les nuages ont en effet tendance 'à atténuer à la fois le refroidissement nocturne et le réchauffement diurne. L'amplitude est de ce fait normalement plus forte en été qu'en hiver.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Dim** | **Lun** | **Mar** |  |  |  | **Jeu** | **Ven** | **Sam** | **Dim** | **Lun** | **Mar** | **Mer** | **Jeu** | **Ven** | **Sam** | **Dim** | **Lun** | **Mar** |
| **Jour** |  | **1** | **2** | **3** |  | **...** |  | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| TN | **0** | **4, 2** | **3, 8** | **4** |  | **...** |  | **7, 6** | **3, 5** | **2, 5** | **2, 2** | **1, 5** | **3, 2** | **2, 1** | ***1, 2*** | ***10, 9*** | **7, 5** | **6, 5** | **6, 2** | **7, 2** |
| TX | **1** | **19, 4** | **20, 3** | **21, 7** |  | **...** |  | **14, 2** | **12, 9** | ***11, 2*** | **13, 7** | **13, 8** | **14, 5** | **17, 4** | **21** | **17, 9** | **19, 8** | **23, 2** | ***23, 8*** | **20, 5** |
| **Précip.** | **2** | **0** | **0** | **0** |  | **...** |  | **4** | **0** | **3** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |

Climatologie à Marrakech (MA) en janvier 2017 source (https://www.infoclimat.fr/climatologie-mensuelle/60230/janvier/2017/marrakech.html)

1. **On peut représenter toutes ces données avec un tableau à deux dimensions Juste/Faux? justifiez!.........................................**

**Définissez les fonctions/procédures suivantes:**

1. **fonction qui retourne le nom du jour *j* en précisant le premier jour *pj* du mois, pj==7 c.à.d. dimanche**

**def jour(pj, j):**

**assert** 1**<=** pj **<=** 7 **and** 1 **<=** j **<=**31

jours **=** **[**'Lundi'**,** 'Mardi'**,** 'Mercredi'**,** 'Jeudi'**,** 'Vendredi'**,** 'Samedi'**,** 'Dimanche'**]**

1. **fonction qui retourne le maximum si max est True, sinon le minimum, pour la mesure donnée par son numéro : 0 → TN 1 → TX 2 → précip.**

**def extremum(meteo, mesure, max):**

**assert 0 <= mesure <= 2**

1. **fonction qui retourne le numéro du *jour* qui correspond à la *valeur* recherchée dans la *mesure* précisée.**

**def recherche(meteo, mesure, valeur):**

1. **fonction qui retourne le tableau de cumul de la *mesure* précisée.**

**def cumul(meteo, mesure):**

1. **fonction qui retourne le numéro du *jour* qui correspond à la valeur maximale d'amplitude.**

**def rechercheMax(meteo):**

1. **procédure qui affiche la date et la valeur de la mesure *meteo* (tab à 1d) précisée pour tous les jours du mois.**

**def afficher(meteo, pj):**

1. **Le programme principal contient le code suivant:**

Tempmin **=** **[**4.2**,** 3.8**,** ...,7.5**,** 6.5**,** 6.2**,** 7.2**]**

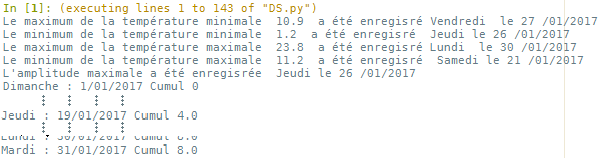
Tempmax **=** **[**19.4**,** 20.3 ...**,** 23.2**,** 23.8**,** 20.5**]**

Precip **=** **[**0**,**..., 0**,** 4.0... 1.0**,** 0**,** 0**]**

meteo **=** **[**Tempmin**,** Tempmax**,** Precip**]**

premierjour **=** 7

mesure **=** 0

**Ecrire le code permettant d'afficher les résultats suivants:**

-------------------------------------------------------------------------------------------------------------------------

***Exercice* :Ecrire une fonction *nb0(n)* qui retourne le nombre de zéros dans un entier. (sans utiliser str()).**

***Problème :*** ---------------------------------------------------------------------------------------------------------

La température minimale (abréviation TN) est la plus basse observée entre 18 heures U.T.C. la veille et 18 heures le jour même.

La température maximale (abréviation TX) est la plus élevée observée entre 6 heures U.T.C. et le lendemain 6 heures. Elle se produit le plus souvent en cours d'après-midi.

La différence entre le minimum et le maximum d'une journée est appelée amplitude diurne. La température minimale est généralement plus élevée lorsque le temps est couvert, et la maximale au contraire plus élevée par temps clair. Les nuages ont en effet tendance 'à atténuer à la fois le refroidissement nocturne et le réchauffement diurne. L'amplitude est de ce fait normalement plus forte en été qu'en hiver.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Dim** | **Lun** | **Mar** |  |  |  | **Jeu** | **Ven** | **Sam** | **Dim** | **Lun** | **Mar** | **Mer** | **Jeu** | **Ven** | **Sam** | **Dim** | **Lun** | **Mar** |
| **Jour** |  | **1** | **2** | **3** |  | **...** |  | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| TN | **0** | **4, 2** | **3, 8** | **4** |  | **...** |  | **7, 6** | **3, 5** | **2, 5** | **2, 2** | **1, 5** | **3, 2** | **2, 1** | ***1, 2*** | ***10, 9*** | **7, 5** | **6, 5** | **6, 2** | **7, 2** |
| TX | **1** | **19, 4** | **20, 3** | **21, 7** |  | **...** |  | **14, 2** | **12, 9** | ***11, 2*** | **13, 7** | **13, 8** | **14, 5** | **17, 4** | **21** | **17, 9** | **19, 8** | **23, 2** | ***23, 8*** | **20, 5** |
| **Précip.** | **2** | **0** | **0** | **0** |  | **...** |  | **4** | **0** | **3** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |

*Climatologie à Marrakech (MA) en janvier 2017 source (https://www.infoclimat.fr/climatologie-mensuelle/60230/janvier/2017/marrakech.html)*

1. **On peut représenter toutes ces données avec un tableau à deux dimensions Juste/Faux? justifiez!**

**Définissez les fonctions/procédures suivantes:**

1. **fonction qui retourne le nom du jour *j* en précisant le premier jour *pj* du mois, pj==7 c.à.d. dimanche**

**def jour(pj, j):**

**assert** 1**<=** pj **<=** 7 **and** 1 **<=** j **<=**31

jours **=** **[**

'Lundi'**,** 'Mardi'**,** 'Mercredi'**,** 'Jeudi'**,**

'Vendredi'**,** 'Samedi'**,** 'Dimanche'

**]**

1. **fonction qui retourne le maximum si max est True, sinon le minimum, pour la mesure donnée.**

**def extremum(mesure, max):**

1. **fonction qui retourne le numéro du *jour* qui correspond à la *valeur* recherchée dans la *mesure* précisée.**

**def recherche(mesure, valeur):**

1. **fonction qui retourne le tableau de cumul de la *mesure* précisée.**

**def cumul(mesure):**

1. **fonction qui retourne le numéro du *jour* qui correspond à la valeur maximale d'amplitude.**

**def rechercheMax(mesure1, mesure0):**

1. **procédure qui affiche la date et la valeur de la mesure précisée pour tous les jours du mois.**

**def afficher(mesure, pj):**

1. **Le programme principal contient le code suivant:**

Tempmin **=** **[**4.2**,** 3.8**,** ...,7.5**,** 6.5**,** 6.2**,** 7.2**]**

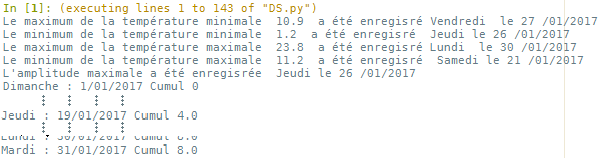
Tempmax **=** **[**19.4**,** 20.3 ...**,** 23.2**,** 23.8**,** 20.5**]**

Precip **=** **[**0**,**..., 0**,** 4.0... 1.0**,** 0**,** 0**]**

premierjour **=** 7

mesure0 **=** Tempmin

mesure1 = Tempmax

**Ecrire le code permettant d'afficher les résultats suivants:**

**Exercice1**

**Que donnent les instructions suivantes?**

**1 - [1, 2, 3] \* 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **A-[3, 6, 9]** | **B-[1, 2, 3] [1, 2, 3] [1, 2, 3]** | **C-[1, 2, 3, 1, 2, 3, 1, 2, 3]** | **D-Erreur** |

**2 - [1, 2, 3] + 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **A - [4, 5, 6]** | **B - [1, 2, 3 , 3]** | **C - [13, 23, 33]** | **D - Erreur** |

**3 - [1, 2, 3] + [3]**

|  |  |  |  |
| --- | --- | --- | --- |
| **A - [4, 5, 6]** | **B - [1, 2, 3 , [3]]** | **C - [1, 2, 3, 3]** | **D - Erreur** |

**3 - [1 for i in range(10) if i % 3 == 0]**

|  |  |  |  |
| --- | --- | --- | --- |
| **A - [1, 1, 1, 1]** | **B - [0, 3, 6, 9]** | **C - [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]** | **D - Erreur** |

**Exercice2**

|  |
| --- |
| 0 \* 9 + 1 = 1 |
| 1 \* 9 + 2 = 11 |
| 12 \* 9 + 3 = 111 |
| 123 \* 9 + 4 = 1111 |
| 1234 \* 9 + 5 = 11111 |
| 12345 \* 9 + 6 = 111111 |
| 123456 \* 9 + 7 = 1111111 |
| 1234567 \* 9 + 8 = 11111111 |
| 12345678 \* 9 + 9 = 111111111 |

**Ecrire une fonction qui retourne True si un entier passé en argument est égal à la somme des puissances 3, 4 ou 5 des chiffres qui le composent.**

**ex.** 3\*\*3 + 7\*\*3 + 1\*\*3 = 371

**def** armstrong**(**nombre**)** **->** bool**:**

**Exercice3**

**Ecrire une fonction qui compte le nombre d'occurrences d'un chiffre dans un entier (sans le convertir en chaîne de caractères.**

**def** compter**(**nombre**,** chiffre**)->** int**:**

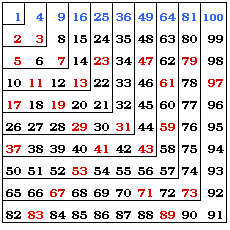
**Exercice4**

**Ecrire un programme qui permet d'afficher le résultat ci-contre en utilisant la boucle for (sans utiliser str()).**

Figure

**PROBLEME:**

**On se propose d'écrire un programme** qui remplit**le tableau T par N = 100 entiers (cf. )**

****

1. Ecrire une fonction qui retourne True si N est le carré d'un entier N = a \* a

**def** carreParfait**(**N**)** **->** bool**:**

1. Ecrire une procédure qui affiche tous les nombres entre 1 et N

**def** tousLesNombres**(**N**):**

1. Ecrire une procédure qui affiche les valeurs d'une colonne j à partir de la diagonale.

**def** ligneV**(**j**):**

Figure tableau T de 100 entiers

1. Ecrire une procédure qui affiche les valeurs d'une ligne i jusqu'à la diagonale.

**def** ligneH**(**i**):**

1. Ecrire une procédure qui permet de remplir un tableau comme dans l'illustration (cf. Figure 2 tableau T de 100 entiers).

**def** remplissage**(**tab**):**

n**,** m **=** len**(**tab**)**

N **=** n **\*** m

**assert** n **==** m

1. Ecrire une procédure qui permet de remplir un tableau comme dans l'illustration (cf. Figure 2 tableau T de 100 entiers). en combinant les codes des procédures précédentes (sans faire appel à ces procédures). en utilisant la boucle while.

**def** remplissage2**(**tab**):**

n**,** m **=** len**(**tab**)**

N **=** n **\*** m

**assert** n **==** m

**Exercice1**

**Que donnent les instructions suivantes?**

**1 - [1, 2, 3] \* 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **A-[3, 6, 9]** | **B-[1, 2, 3] [1, 2, 3] [1, 2, 3]** | **C-[1, 2, 3, 1, 2, 3, 1, 2, 3]** | **D-Erreur** |

**2 - [1, 2, 3] + 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **A - [4, 5, 6]** | **B - [1, 2, 3 , 3]** | **C - [13, 23, 33]** | **D - Erreur** |

**3 - [1, 2, 3] + [3]**

|  |  |  |  |
| --- | --- | --- | --- |
| **A - [4, 5, 6]** | **B - [1, 2, 3 , [3]]** | **C - [1, 2, 3, 3]** | **D - Erreur** |

**3 - [1 for i in range(10) if i % 3 == 0]**

|  |  |  |  |
| --- | --- | --- | --- |
| **A - [1, 1, 1, 1]** | **B - [0, 3, 6, 9]** | **C - [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]** | **D - Erreur** |

**Exercice2**

|  |
| --- |
| 0 \* 9 + 8 = 8 |
| 9 \* 9 + 7 = 88 |
| 98 \* 9 + 6 = 888 |
| 987 \* 9 + 5 = 8888 |
| 9876 \* 9 + 4 = 88888 |
| 98765 \* 9 + 3 = 888888 |
| 987654 \* 9 + 2 = 8888888 |
| 9876543 \* 9 + 1 = 88888888 |
| 98765432 \* 9 + 0 = 888888888 |

**Ecrire un programme qui permet d'afficher le résultat ci-contre en utilisant la boucle for.**

**Exercice3**

**Un nombre parfait est un nombre N qui est égal à la somme de ses diviseurs dvs propres (qui sont inférieurs strictement à N).**

**Ecrire une fonction qui retourne True si le nombre passé en argument est un nombre parfait, ex. 1 + 2 + 3 = 6**

**def** parait**(**N**)** **->** bool**:**

**Exercice4**

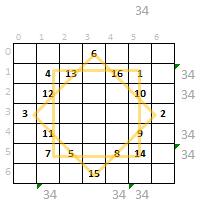
**Ecrire un programme qui permet d'afficher tous les nombre parfaits inférieurs strictement à un entier n**

Figure

**Problème**

**On se propose d'écrire un programme qui vérifie si un tableau T à deux dimensions, représente une étoile magique, c.à.d. étoile dont la somme des valeurs des sommets et intersections est la même pour chaque segment.**

**Les autre cases sont initialisées toutes à 0.**

1. Ecrire une fonction qui retourne la somme des éléments de la colonne j, entre les lignes id et if.

**def** sommeColone**(**j, i\_d, i\_f**)** **->** int**:**

1. Ecrire une fonction qui retourne la somme des éléments de la ligne i, entre les colonnes j\_d et j\_f.

**def** sommeLigne **(**i, j\_d, j\_f**):**

1. Ecrire une fonction qui retourne la somme des éléments de la diagonale entre (i\_d,j\_d), et (i\_f,j\_f), avec:

* i\_d ligne de début

Figure étoile magique

* j\_d colonne de début
* i\_f ligne de fin
* j\_f colonne de fin

**def** diagonale**(**id**,** j\_d, i\_f**,** j\_f**):**

1. Ecrire une fonction retourne True si le tableau T représente une étoile magique comme celle présentée dans l'illustration.

**def** etoileMagique**(**T**)** **->** bool**:**

1. Ecrire une fonction qui retourne la somme des éléments d'un segment entre (i\_d,j\_d), et (i\_f,j\_f), avec:

* i\_d ligne de début
* j\_d colonne de début
* i\_f ligne de fin
* j\_f colonne de fin

***(\*sans faire appel aux autres fonctions)***

**def** sommeSegment**(**id**,** j\_d, i\_f**,** j\_f**):**