|  |  |  |
| --- | --- | --- |
| Spécialité NSI Première | **DEVOIR SURVEILLE DE** | Lundi 15 mai 2023 |
| Lycée d’Avesnières | **NSI** | Durée : 55 mn |
| Année scolaire 2022-2023 | **N° 7** | Calculatrice interdite |

**NOM : ..........................................................**

**Prénom : ......................................................**

**Complétez directement sur l’énoncé** les programmes en langage Python.

# Agence de location

Une agence de location de motos à deux roues loue ses motos à 100 € la journée. Il faut ajouter 0,50 € par km pour les 50 premiers kilomètres puis 0,70 € par km au-delà.

Compléter le code Python de la fonction prix\_location(nombre\_km) prenant en argument :

* un entier nombre\_km

et renvoyant un flottant  prix, qui est le prix à payer par le client ayant loué pendant une journée et ayant parcouru la distance  nombre\_km.

***Exemples***

Si nombre\_km = 10

alors prix\_location(nombre\_km) renvoie 105.0

Si nombre\_km = 55

alors prix\_location(nombre\_km) renvoie 128.5

|  |
| --- |
| def prix\_location(nombre\_km):        if nombre\_km ...          prix = ...        else:          prix\_premiers\_50km = ...          nombre\_km\_restants = ...          prix\_au\_dela\_50km = ...            prix = ...        return prix |

# Somme

Écrire une fonction somme prenant en argument :

* une liste  L de nombres entiers ou flottants

et renvoyant la variable total de type entier ou flottant et qui vaut la somme des éléments de cette liste.

On interdit ici d’utiliser sum().

***Exemple***

Si L = [8, 12, 4, 1, 0, -5]

alors somme(L) renvoie 20

|  |
| --- |
| def somme(L):          return total |

# Comptage de mots

Compléter la fonction comptage\_de\_mots(phrase) prenant en argument :

* Une chaîne de caractères nommée phrase

et renvoyant le nombre de mots dans la phrase passée en argument.

***Exemples***

Si phrase = "chat"

alors comptage\_de\_mots(phrase) renvoie 1

Si phrase = "Le chat noir dort"

alors comptage\_de\_mots(phrase) renvoie 4

Si phrase = "Le chat noir dort sur l'étagère"

alors comptage\_de\_mots(phrase) renvoie 6

|  |
| --- |
| def comptage\_de\_mots(phrase):      nombre\_de\_mots = ...      for lettre in phrase:          if ...              nombre\_de\_mots = nombre\_de\_mots + 1      return nombre\_de\_mots |

# Maximum

Écrire une fonction maximum prenant en argument :

* une liste non vide  L de nombres entiers

et renvoyant la variable maxi de type entier et qui vaut le plus grand entier présent dans cette liste.

On interdit ici d’utiliser max().

***Exemple***

Si L = [8, 12, 4, 1, 0, -5]

alors maximum(L) renvoie 12

|  |
| --- |
| def maximum(L):          return maxi |

# Résultats d’un vote

Lors d’un vote, il y a trois candidats ‘A’, ‘B’ et ‘C’.

Dix personnes ont déposé leur bulletin dans l’urne. Celle-ci est représentée par un tableau.

Exemple : urne\_1 = ['A', 'A', 'A', 'B', 'C', 'B', 'C', 'B', 'C', 'B']

La fonction depouille(urne) doit permettre de compter le nombre de votes exprimés pour chacun des candidats. Elle prend en paramètre un tableau et renvoie le résultat dans un dictionnaire dont les clés sont les noms des candidats et les valeurs le nombre de votes en leur faveur.

Compléter la fonction depouille pour qu’elle renvoie le résultat attendu.

***Exemple***

depouille(urne\_1) renvoie {'A': 3, 'B': 4, 'C': 3}.

|  |
| --- |
| def depouille(urne):      resultat = ...      for bulletin in urne:          if ...              resultat[bulletin] = resultat[bulletin] + 1          else:              ...      return resultat |

# Détecter un tableau trié

Écrire une fonction detecte prenant en argument :

* une liste  L de nombres entiers ou flottants

et renvoyant la variable est\_trie de type booléen et qui vaut True lorsque tous les éléments de cette liste sont déjà triés par ordre croissant et False sinon.

***Exemple***

Si L = [2, 12, 14, 14, 40, 5]

alors detecte(L) renvoie False

Si L = [2, 12, 14, 14, 40, 52]

alors detecte(L) renvoie True

|  |
| --- |
| def detecte(L):          return est\_trie |

# Tri par sélection

Écrire une fonction tri\_selection prenant en paramètre une liste tab de nombres entiers et qui renvoie la liste triée par ordre croissant. Il est demandé de ne pas créer de nouvelle liste mais de modifier celle fournie.

On utilisera l’algorithme suivant :

* On recherche le plus petit élément de la liste, en la parcourant du rang 0 au dernier rang, et on l’échange avec l’élément d’indice 0 ;
* On recherche ensuite le plus petit élément de la liste restreinte du rang 1 au dernier rang, et on l’échange avec l’élément d’indice 1 ;
* On recherche ensuite le plus petit élément de la liste restreinte du rang 2 au dernier rang, et on l’échange avec l’élément d’indice 2 ;
* …
* On continue de cette façon jusqu’à ce que la liste soit entièrement triée.

***Exemples***

Si tableau = [1, 52, 6, -9, 12]

alors tri\_selection(tableau) renvoie [-9, 1, 6, 12, 52]

|  |
| --- |
| def tri\_selection(tab):      for i in range(...          indice\_mini = i          for j in range(...              if ... < ...                  indice\_mini = ...          tab[i], tab[indice\_mini] = ...      return tab |

# Conversion

On modélise la représentation binaire d’un entier non signé par un tableau d’entiers dont les éléments sont 0 ou 1. Par exemple le tableau [1, 0, 1, 0, 0, 1, 1] représente l’écriture binaire de l’entier dont l’écriture décimale est

1\*(2\*\*6) + 0\*(2\*\*5) + 1\*(2\*\*4) + 0\*(2\*\*3) + 0\*(2\*\*2) + 1\*(2\*\*1) + 1\*(2\*\*0) = 83.

A l’aide d’un parcours séquentiel du tableau, écrire la fonction convertir répondant aux spécifications suivantes :

|  |
| --- |
| def convertir(tab):      """        Parameters      ----------      tab : TYPE : tableau d'entiers          Ses éléments sont 0 ou 1 et représentent un entier écrit en binaire.      Returns      -------      ecriture\_decimale : TYPE : entier          C'est l'écriture décimale de l'entier positif dont la représentation          binaire est donnée par le tableau tab.      """                    return ecriture\_decimale |

***Exemples***

Si tableau = [1, 0, 1, 0, 0, 1, 1]

alors convertir(tableau) renvoie 83

Si tableau = [1, 0, 0, 0, 0, 0, 1, 0]

alors convertir(tableau) renvoie 130