

Nom :	Prénom :	Classe :
-------	----------	----------

## NSI 1re — DS 05

Coefficient 3

Note ramenée à 20 sur Pronote

Calculatrice interdite

### 1) Documenter une fonction — 4 pts

Documenter les fonctions ci-dessous.

N°	Code Python	Documentation (à compléter)
1	<pre>def a(b):     """....."""     return b + 1</pre>	<p>""""Renvoie le résultat de “b” additionné de 1""""</p>
2	<pre>def c(d, e):     """....."""     return d / e</pre>	<p>""""Renvoie le résultat (flottant) de “d” divisé par “e”""""</p>
3	<pre>def f(g, h):     """.....     .....     .....     .....     """     i = g // h     j = g % h     return (i, j)</pre>	<p>""""Renvoie le <b>tuple</b> contenant le quotient (résultat entier d'une division) et le reste de la division de “g” par “h”.</p> <p>Paramètres :</p> <ul style="list-style-type: none"> <li>- g (int / float)</li> <li>- h (int / float)</li> </ul> <p>""""</p>
4	<pre>def minutes(h, m, s):     """.....     .....     .....     .....     """     r = h * 3600     r += m * 60     r += s     return r</pre>	<p>""""Renvoie le résultat correspondant au nombre de secondes contenue dans l'heure formée par “h”, “m” et “s”.</p> <p>Paramètres :</p> <ul style="list-style-type: none"> <li>- h (int) : heures</li> <li>- m (int) : minutes</li> <li>- s (int) : secondes</li> </ul> <p>""""</p>

## 2) Écrire une fonction à partir de sa documentation — 4 pts

Écrire les fonctions correspondants aux documentations ci-dessous.

N°	Documentation	Code Python (à compléter)
1	<pre>"""Calcule et renvoie le volume d'un cube, à partir de la formule C³ (où C est la longueur d'une arête du cube).  Paramètre : c (float): Le côté d'un carré. """</pre>	<pre>def f (c):     return c *** 3</pre>
2	<pre>"""Calcule et renvoie l'aire d'une sphère, à partir de la formule 4 × π × R² (où R est le rayon de la sphère).  Paramètres : r (float): Rayon de la sphère. pi (float): Valeur de π. """</pre>	<pre>def f(r, pi):     return 4 * pi * r ** 2</pre>

La suite page suivante →

N°	Documentation	Code Python (à compléter)
3	<p>"""Permet de savoir si une conductrice ou un conducteur roule trop vite, ou non, sur autoroute.</p> <p>Paramètres :</p> <p>v (int): La vitesse (en km/h).</p> <p>Renvoie :</p> <p>'Danger' (str) si la vitesse est supérieure à 130</p> <p>'OK' (str) sinon</p> <p>"""</p>	<pre>def f(v):     if v &gt; 130:         return 'Danger'     else:         return 'OK'</pre>
4	<p>"""Calcule la moyenne, la note minimale et la note maximale d'un élève, pour un ensemble de notes donné.</p> <p>Paramètres :</p> <p>notes (list): Les notes obtenues par l'élève.</p> <p>Renvoie :</p> <p>(tuple) contenant 3 valeurs : moyenne, note mini, note maxi</p> <p>"""</p>	<pre>def f(notes):     min = notes[0]     max = notes[0]     accu = 0     comp = 0     for v in notes:         accu += v         comp += 1         if v &lt; min:             min = v         if v &gt; max:             max = v     moy = accu / comp     return (moy, min, max)</pre>

### 3) Écrire une fonction à partir d'un problème donné — 5 pts

---

Un hôtel propose les tarifs suivants :

- Tarif standard : 150 EUR par nuit.
- Promo à partir de 3 nuits : 135 EUR par nuit.
- Promo à partir de 5 nuits : 120 EUR par nuit.
- Promo à partir de 7 nuits : 100 EUR par nuit.

Option petit déjeuner : +15 EUR par nuit.

Quelle **fonction** permettant de calculer le **coût total** proposeriez-vous à cet établissement ?

```
1 def f(nuits, dej):
2     t = 150
3     if nuits >= 7:
4         t = 100
5     elif nuits >= 5:
6         t = 120
7     elif nuits >= 3:
8         t = 135
9     if dej == True:
10        return t * nuits + nuits * 15
11    else:
12        return t * nuits
```

#### 4) Identifier un algorithme — 3 pts

Identifier les algorithmes ci-dessous, en les nommant convenablement.

N°	Code Python	Nom de l'algorithme
1	<pre>for i in range(1, len(tableau)):     cle = tableau[i]     j = i     while tableau[j-1] &gt; cle and j &gt; 0:         tableau[j] = tableau[j-1]         j = j - 1     tableau[j] = cle</pre>	Tri par insertion
2	<pre>for i in range(len(tableau) - 1):     valeur_min = tableau[i]     indice_min = i     for j in range(i + 1, len(tableau)):         valeur_observed = tableau[j]         if valeur_observed &lt; valeur_min:             indice_min = j             valeur_min = tableau[j]     tableau[indice_min] = tableau[i]     tableau[i] = valeur_min</pre>	Tri par sélection
3	<pre>i_g = 0 i_d = len(tableau) - 1 while i_g &lt;= i_d:     i_m = (i_g + i_d) // 2     v_m = tableau[i_m]     if v_cherchee &lt; v_m:         i_d = i_m - 1     elif v_cherchee &gt; v_m:         i_g = i_m + 1     else:         print(i_m)         break</pre>	Recherche dichotomique

## 5) Écrire un algorithme élémentaire sur un tableau indexé — 4 pts

a) Créer le programme permettant de calculer la **moyenne** des éléments présents dans un tableau quelconque. *Pour rappel, une moyenne se calcule en divisant la somme des valeurs par son nombre.* Attention : les fonctions len et sum sont interdites.

```
1 acc = 0
2 com = 0
3 for v in tab:
4     acc += v
5     com += 1
6 print(acc / com) # Moyenne
```

b) Créer le programme permettant de compter le **nombre d'occurrences** d'une valeur dans un tableau quelconque. Attention : la fonction count est interdite.

```
1 occ = 0
2 for v in tab:
3     if v == 389: # Où 329 est la valeur recherchée
4         occ += 1
5 print('La valeur apparaît', occ, 'fois')
```

La suite page suivante →

c) Créer le programme permettant de chercher la **valeur minimale** présente dans un tableau quelconque. Attention : la fonction min est interdite.

```
1 min = tab[0]
2 for v in tab:
3     if v < min:
4         min = v
5 print(min)
```

d) Créer le programme permettant de rechercher **l'indice** d'une valeur présente dans un tableau quelconque. On suppose la valeur spécifique comme étant unique. Attention : la fonction index est interdite.

```
1 i_trouve = False
2 for i in range(len(tab)):
3     if tab[i] = 389: # Où 329 est la valeur recherchée
4         i_trouve = i
5 print(i_trouve)
```