	<p align="center">Institut Supérieur des Etudes Technologiques de Sousse</p> <p align="center">Département de Technologies de l'Informatique Licence Appliquée en Technologies de l'Informatique</p> <p align="center"><b>EXAMEN FINAL</b></p>	<p><b>Année Universitaire :</b> 2023/2024</p> <p><b>Semestre :</b> 1</p> <p><b>Date:</b> 04 Janvier 2024 <b>Durée :</b> 1h30</p>
<p><b>Documents :</b> Non autorisés</p>	<p align="center"><b>Unité d'enseignement :</b> Optionnel</p> <p align="center"><b>Matière :</b> Préparation à la Certification Python 1</p>	<p align="center"><b>Classe :</b> TI1.* <b>Nb. Pages:</b> 04</p>
<p align="center"><b>Enseignants :</b> M<sup>me</sup> Hédia Jegham, M<sup>me</sup> Ahlem El Haj, M<sup>r</sup> Ghazi Ben Jdidia, M<sup>r</sup> Nizar Rouatbi, Mr Yassine Kachoukh</p>		

✓ **Propreté de la feuille d'examen et clarté du code sont exigées !**

### Exercice 1 QCM (5 pts) (0.5 X 10)

*Reportez sur votre feuille de réponse le numéro de la question et la/les bonne(s) réponse(s).*

<p><b>1. What is the output of the following snippet?</b></p> <pre>dct = {} dct['1'] = (1, 2) dct['2'] = (2, 1) for x in dct.keys():     print(dct[x][1], end='')</pre>	<p><b>2. What is the expected output of the following code?</b></p> <pre>x = [0, 1, 2] x.insert(0, 1) del x[1] print(sum(x))</pre>
<p>[1] 21 [2] (2,1) [3] (1,2) [4] 12</p>	<p>[1] 3 [2] 2 [3] 4 [4] 5</p>
<p><b>3. What is the output of the following snippet?</b></p> <pre>tup=(1,)+(1,) tup=tup+tup print(len(tup))</pre>	<p><b>4. What is the expected output of the following code?</b></p> <pre>data = ['Hi', 404, 3.03, 'Ok', 33.3] print(data[1:3])</pre>
<p>[1] The snippet is erroneous (invalid syntax) [2] 2 [3] 4 [4] 4,</p>	<p>[1] [404, 3.03, 'Ok']. [2] ['Hi', 404, 3.03, 'Ok', 33.3] [3] ['Hi', 'Ok'] [4] [404, 3.03]</p>
<p><b>5. What is the output of the following snippet?</b></p> <pre>dictionary={'one':'two', 'three':'one', 'two':'three'} v=dictionary['one'] for k in range(len(dictionary)):     v=dictionary[v] print(v)</pre>	<p><b>6. What will be the output of the following code snippet?</b></p> <pre>d = {} d[1] = 1 d['1'] = 2 d[1] += 1 s = 0 for k in d:     s += d[k] print(s)</pre>
<p>[1] one [2] two [3] ('one', 'two', 'three') [1] three</p>	<p>[1] 4 [2] 3 [3] 2 [4] 1</p>

<b>7. What is the expected output of the following code?</b> <pre>print(list('hello'))</pre>	<b>8. What is the default return value for a function that does not explicitly return any value?</b>
<pre>[1] ['hello'] [2] [h, e, l, l, o] [3] ['h', 'e', 'l', 'l', 'o'] [4] ['h' 'e' 'l' 'l' 'o']</pre>	<pre>[1] int [2] None [3] void [4] Null</pre>
<b>9. Assuming this tuple t = (1, 2, 4, 4) which one of these instructions causes an error? (choose two answers)</b>	<b>10. What is the expected output of the following code?</b> <pre>my_dict={1:'a', ([1,2],3):'b'} print(my_dict[1])</pre>
<pre>[1] t[2] = 3 [2] print(t[2]) [3] t.append(10) [4] print(len(t))</pre>	<pre>[1] a [2] 1 [3] A TypeError exception [4] b</pre>

## Exercice 2 Tournage à la main (4 pts)

Soit le Python script suivant :

```
def calcul(n=1):
    ld = []
    for i in range(1, n + 1):
        if n % i == 0 : ld.append(i)
    return ld
print(calcul(12))
print(calcul())
```

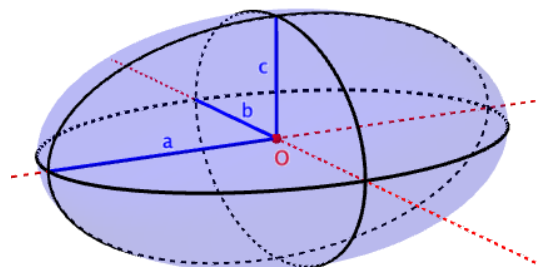
1. Indiquez l'affichage obtenu suite à l'exécution de ce script.
2. Expliquez brièvement le rôle de la fonction calcul()

## Problème Programmation Python (11 pts)

Une société d'import/export compte ranger des marchandises qui sont sous forme d'ellipsoïdes dans des conteneurs de sorte à faire le minimum de gaspillage. On donne dans ce qui suit la figure d'un ellipsoïde.

La forme tridimensionnelle d'un ovale est appelée « ellipsoïde ». Un ellipsoïde est une figure géométrique 3D qui ressemble à un ovale allongé ou aplati. Il peut avoir des axes de longueur différente, ce qui le distingue d'une sphère où tous les axes ont la même longueur.

Le volume d'un ellipsoïde est donné par la formule suivante :



$$\frac{4}{3} \pi \times a \times b \times c$$

- [1] Ecrire une fonction `Saisie_Vals()` qui permet à l'utilisateur de saisir **3 réels strictement positifs** puis de retourner **la liste** contenant ces 3 réels triés dans **l'ordre décroissant** (du plus grand au plus petit).

[2] Ecrire le script d'une fonction `Axes_ellipsoide ()` qui fait appel à la fonction de la question précédente `Saisie_Vals ()` pour assister l'utilisateur à saisir les dimensions a, b, et c d'un ellipsoïde général puis les remettre **sous la forme d'un tuple (a,b,c)**. Ces dimensions sont interprétées comme suit : a est la longueur du demi-axe principal, b est la longueur du demi-axe intermédiaire, et c, est la longueur du demi-axe mineur, ce qui signifie que **a>b>c**.

La fonction doit assurer à travers **un contrôle de saisie** que les 3 dimensions a, b et c sont **différentes deux à deux** pour éviter que l'ellipsoïde ne se réduise à un galet, à un ballon de rugby ou à une sphère. La fonction devra afficher un message explicatif (indiquant une erreur de saisie) si l'un des trois cas suivants subvient :

Si  $a = b = c$ , message : l'ellipsoïde est une sphère.

Si  $a = b$ , message : l'ellipsoïde aplati a une forme de galet.

Si  $b = c$ , message : l'ellipsoïde allongé a une forme de ballon de rugby.

[3] Ecrire le script d'une fonction `Vol_ellipsoide ()` qui reçoit trois paramètres a,b,c et remet le volume d'un ellipsoïde (voir formule sous la figure de l' ellipsoïde).

[4] Ecrire une fonction `Qte_ellipsoide()` qui utilise les deux fonction précédentes (`Axes_ellipsoide` et `Vol_ellipsoide`) pour assister l'utilisateur à **saisir les dimensions** d'une marchandise E qui a la forme d'ellipsoïde général, **calculer son volume** ; ensuite **saisir sa quantité q (entier strictement positif)**. La fonction renvoie alors **une liste M** qui a la structure suivante :

$M : [(volume, q), (a, b, c)]$

Le premier élément de la liste M est un tuple contenant le volume et la quantité (nombre de pièces) de la marchandise, alors que le deuxième élément est un tuple indiquant les dimensions de la marchandise. Voici un exemple de liste renvoyée par cette fonction : [(1.01, 200), (1.5, 0.8, 0.2)]

[5] Ecrire une fonction `Saisie_Marchandises ()` qui fait appel à la fonction de la question précédente `Qte_ellipsoide()` pour assister l'utilisateur à saisir les informations (noms, dimensions et quantités) de **deux marchandises différentes** E1 et E2 qui ont la forme d'ellipsoïdes généraux, et calculer leurs volumes.

La fonction remettra un dictionnaire `dicoE` qui a la structure suivante :

```
dicoE={E1 :[(volume1, q1), (a1,b1,c1)], E2 :[(volume2, q2), (a2,b2,c2)] }
```

Les Ei sont des clés de type str, et correspondent aux noms des marchandises saisies par l'utilisateur.

Les valeurs du dictionnaire sont des listes de tuples contenant des informations sur la marchandise correspondante (volume, quantité et dimensions). Voici un exemple de dictionnaire renvoyé par cette fonction :

```
{'Alyassine': [(1.01, 200), (1.5, 0.8, 0.2)], ' AlKassem': [(0.44, 150), (0.7, 0.5, 0.3)]}
```

Dans cet exemple, la marchandise de nom 'Alyassine' a pour dimensions en mètre (1.5, 0.8, 0.2) mètre, un volume de 1.01 m<sup>3</sup> et il en existe 200 pièces de cette marchandise.

[6] Notre objectif est d'aider la société à ranger les deux types de marchandises E1 et E2 dans des conteneurs de sorte à faire le minimum de gaspillage. Sachant que les conteneurs sont tous **de volume égal** et que chaque conteneur ne peut contenir **qu'un seul type de marchandise**, écrire une fonction `repartir_E ()` qui reçoit en argument le dictionnaire `dicoE` dont la structure est décrite précédemment, ainsi que le volume `volCont` d'un conteneur, et qui calcule et renvoie un tuple constitué du nombre de conteneurs nécessaire pour la première marchandise et du nombre de conteneurs nécessaire pour la deuxième marchandise.

**Indication :** Le nombre de conteneurs doit être un entier. Vous pouvez alors utiliser la fonction `math.ceil()` de la bibliothèque `math`. C'est une fonction qui arrondit un nombre à l'entier supérieur : elle renvoie le plus petit entier supérieur ou égal au nombre donné en entrée. Par exemple, `math.ceil(3.2)` renvoie 4, et `math.ceil(4)` renvoie 4 aussi.

[7] Sachant que la douane offre des conteneurs ayant chacun un volume utile de **40 m<sup>3</sup>**, écrire le script python du programme principal qui permet de saisir le dictionnaire `dicoE` contenant les informations sur les deux marchandises puis affiche le nombre de conteneurs nécessaires pour stocker l'ensemble des marchandises.

Exemple pour ce dictionnaire de marchandise :

```
{'Alyassine': [(1.01, 200), (1.5, 0.8, 0.2)], 'AlKassem': [(0.44, 150), (0.7, 0.5, 0.3)]}
```

le programme remet :

Le nombre de conteneurs pour première et deuxième marchandise : (6, 2)

*Bon travail*