TP4 :
Boucles

Exercice 1 : **Somme de nombres**

Q1 : En utilisant une boucle for, calculer la somme des entiers positifs inférieurs ou égaux à 1000.

Q2 : Modifier le programme pour réaliser la somme des entiers positifs inférieurs ou égaux à 1000 qui ne sont divisibles ni par 3 ni par 5.

Exercice 2 : **Diviseurs**

Q1 : Ecrire un programme qui demande à l'utilisateur de saisir un nombre entier strictement positif, et qui affiche successivement ses diviseurs (entiers strictement positifs).

Q2 : Modifier le programme pour qu'il affiche également le nombre de diviseurs de cet entier.

Exercice 3 : **Nombre premier**

On souhaite un programme qui demande à l'utilisateur de saisir un nombre entier strictement positif, et qui affiche True si ce nombre est premier, False sinon.

Q1 : Ecrire ce programme.

Listes

Exercice 4 : **Manipulation de listes**

Une liste est définie entre crochets, avec un séparateur entre les éléments : la virgule.

```
maliste = [’Lun’, ’mar’, ’Sam’, ’ Dim’]
```

Q1 : Définir la liste `maliste` en mémoire.

Q2 : Dans un programme, à l'aide d'une boucle, construire une nouvelle liste `newliste`, de même contenu que `maliste`, mais d'ordre inversée. (attention, il est parfois nécessaire de définir une nouvelle variable de liste avant de l'utiliser. Par exemple : `newliste=[]`).

Exercice 5 : **Recherche d'intrus**

On considère la liste hétérogène ['lun','mar',3,'jeu','ven','sam','dim']

Q1 : Proposer un programme qui, à l'aide d'une boucle, recherche l'intrus (élément de type différent) dans une liste sachant que l'intrus est unique et qu'il n'est pas en première position. Faire afficher l'index de l'intrus

Q2 : Compléter le programme pour qu'il affiche la liste dont il aura retiré l'intrus.

Arrays

Exercice 6 : **Tracé de courbe : comparaison de syntaxes**

Les trois programmes ci-dessous permettent de tracer la même courbe, avec des outils et des syntaxes différentes.

```
from math import cos,pi
from matplotlib import pyplot as plt
for x in range(11):
    y=2*cos(pi*2*x/10)
    plt.plot(x/10,y,'+b')
```

```
from math import cos,pi
from matplotlib import pyplot as plt
X=[t/10 for t in range(11)]
Y=[2*cos(pi*2*x) for x in X]
plt.plot(X,Y)
```

```
from numpy import array,arange,cos,pi
from matplotlib import pyplot as plt
X=arange(0,1.1,0.1)
Y=2*cos(2*pi*X)
plt.plot(X,Y)
```

Q1 : Expliquer les différences entre chaque programme.

Q2 : Quelle est la complexité algorithmique de chaque programme, en fonction de n le nombre de points de la courbe.

Tracés

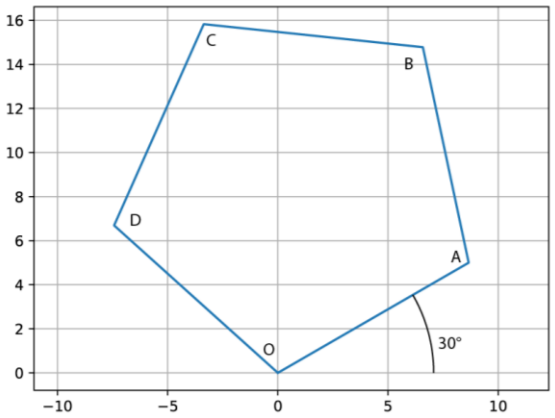
Exercice 7 : **Tracés de polygones**

Dans un repère orthonormé $(O, \vec{x}, \vec{y})$:

- Donner les coordonnées des 4 sommets d'un carré de côté 50, dont un sommet est confondu avec l'origine O, et dont tous les points ont des coordonnées positives ou nulles.
- Tracer ce carré en utilisant `matplotlib`.
- Tracer l'image de ce carré par une translation de vecteur $(25,10)$. On déterminera au préalable les coordonnées des nouveaux sommets.
- Tracer l'image du premier carré par une rotation de centre O et d'angle $+30°$. On déterminera au préalable les coordonnées des nouveaux sommets.

Exercice 8 : **Tracé de pentagone**

Écrire programme permettant de tracer le pentagone régulier comme ci-contre. Ses côtés ont une longueur 10. On pourra utiliser les fonctions `cos` et `sin` après les avoir importées en mémoire, ainsi que la constante `pi`.




```
from numpy import cos,sin,pi
```