

Exercice 1 (Manipulation de nombres...) On donne le script suivant :

```
def ordonne() :  
    a=int(input("Votre premier nombre ? "))  
    b=input("Votre second nombre ? ")  
    if a < b  
        return a, b  
    else  
        b, a
```

1. Corrigez les erreurs qui se sont glissées dans ce script.
2. Tapez le script corrigé, enregistrez le dans un fichier `nombre.py` et testez le.
3. Expliquez ce que fait ce script.

Exercice 2 (Indice de Masse Corporelle) L'Indice de Masse Corporelle (IMC) est le rapport du poids (exprimée en kg) par le carré de la taille (exprimée en m). Il permet de déterminer la corpulence d'une personne :

- en dessous de 18, on souffre de maigreur ;
- entre 18 et 25, la corpulence est idéale ;
- entre 25 et 30, on est en surpoids ;
- au delà de 30, on souffre d'obésité.

Ecrivez un script donnant l'IMC et la corpulence d'une personne.

La bibliothèque `Turtle` permet de réaliser des dessins géométriques correspondant à la trace d'une petite « tortue » virtuelle (représentée par un petite flèche) se déplaçant dans une fenêtre graphique. Le déplacement de la tortue sur l'écran est contrôlé par des instructions simples issues de la bibliothèque `Turtle`. Pour charger cette bibliothèque, n'oubliez pas de commencer votre script par :

```
from turtle import *
```

Exercice 3 (Un exemple de tracé avec `turtle`)

Tapez le programme suivant et interprétez les lignes de code en fonction de l'affichage.

```
from turtle import *  
  
def ExempleTrace() :  
    setup(900 , 600)  
    up() ; setheading(90) ; goto(0 , -300) ; down()  
    color('yellow')  
    goto(0, 300)  
    up() ; goto(-175, 0) ; down()  
    color('blue', 'blue')  
    begin_fill() ; circle(50) ; end_fill()  
    up() ; goto(275, 0) ; down()  
    color('red', 'green')  
    begin_fill() ; circle(50) ; end_fill()  
    exitonclick()  
  
ExempleTrace()
```

Exercice 4 (Contrôle de distance)

1. Dans un fichier `DistanceTurtle.py` programmer en Python la fonction suivante, testez-la et expliquez à quoi elle sert :

```
from turtle import *  
  
def droite(d) :  
    maxX= window_width()/2  
    x, y = position()  
    if x + d < maxX :  
        goto(x + d, y)
```

2. Ecrivez sur le même modèle une fonction `bas(distance)` déplaçant la tortue verticalement en vérifiant qu'elle ne sort pas de la fenêtre graphique.
3. Programmez la fonction suivante, testez-la et expliquez à quoi elle sert :

```
def horizontal(distance) :  
    if distance > 0 :  
        forward(distance)  
    else :  
        backward(-distance)
```

4. Ecrivez sur le même modèle une fonction `vertical(distance)` déplaçant la tortue verticalement en vérifiant qu'elle ne sort pas de la fenêtre graphique.

Exercice 5 (Encore des nombres...) On donne les deux scripts suivants :

Script 1

```
def calcul1() :  
    for k in range(1,22,2) :  
        print(k)
```

Script 2

```
def calcul2(n) :  
    s=0  
    for k in range(n+1) :  
        s=s+k**3  
    print(s)
```

1. Tapez les deux scripts, enregistrez les dans un fichier `nombre2.py` et testez les.
2. Expliquez ce que font ces scripts.

Exercice 6 (Tables) On donne le script suivant :

```
def table() :  
    for a in range(1,10) :  
        print("Table de "+str(a)+" :")  
        for b in range(1,10) :  
            print(str(a)+"x"+str(b)+"="+str(a*b))  
        print("\n")
```

1. Tapez ce script, enregistrez le dans un fichier `table.py` et testez le.
2. Expliquez ce que fait ce script.

Exercice 7 (Cercle, carré ou triangle?)

1. Complétez la fonction suivante :

```
from turtle import *  
  
def Trace(figure) :  
    '''Trace la figure choisie (un carre, un cercle ou un triangle),  
    la variable figure est une chaine de caractères.  
    Si la chaine de caractères saisie n'est pas "carre" ou "cercle",  
    la figure tracée sera un triangle. '''  
  
    if figure == "cercle" :  
        circle(100)  
    elif figure == "carre" :  
        .....  
    else :  
        .....
```

2. A quoi sert le texte entre `'''` ?

Exercice 8 On donne la fonction `Equi(cote)` suivante. Testez la et expliquez ce qu'elle fait :

```
setup(200,200)
def Equi(cote) :
    for i in range(3) :
        forward(cote)
        right(120)
```

Exercice 9 (Figures géométriques) Dans une fenêtre graphique de taille 600×600 :

1. Ecrivez un programme permettant de tracer un carré de côté 100 avec la tortue.
2. Modifiez le programme précédent pour tracer un pentagone régulier de côté 50.
3. Améliorez encore votre programme en écrivant une fonction `PolygoneRegulier` prenant en entrée le nombre de côté et leur longueur afin de tracer un polygone régulier.

Exercice 10 (Evolution d'un capital 1) On reprend l'exemple du cours avec une somme de 2000 € déposée sur un compte rémunéré au taux annuel de 2%.

1. Comment modifier ce script pour que l'on puisse afficher la somme disponible au bout de n'importe quelle durée en années ?
2. Comment modifier ce script pour qu'il puisse fonctionner avec n'importe quelle somme, n'importe quel taux et n'importe quelle durée en années ?

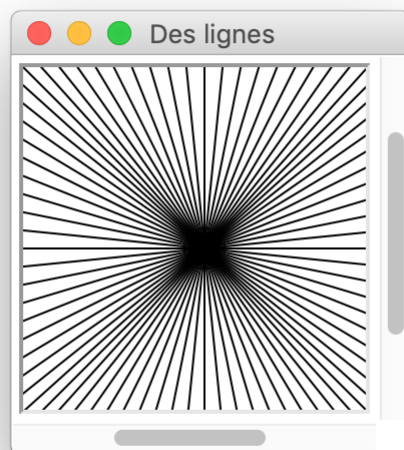
Exercice 11 On donne la fonction Python suivante :

```
from random import randint

def Simulation(n) :
    S = 0
    for k in range(1,n+1) :
        x = randint(1,10)
        if x < 5 :
            S = S + 1
    return S, S/n
```

1. Quelles sont les instructions algorithmiques et les variables utilisées dans cet algorithme ?
2. Ecrire en langage naturel l'algorithme correspondant à script.
3. On saisit `Effectif, Frequence = Simulation(1000)`. Quelles sont les valeurs stockées dans les variables `Effectif` et `Frequence` ?

Exercice 12 (Des lignes) Ecrivez un programme permettant de réaliser, dans une fenêtre de taille 200×200 , le dessin suivant :



Modifiez ensuite ce programme pour qu'il puisse fonctionner avec n'importe quelle fenêtre graphique

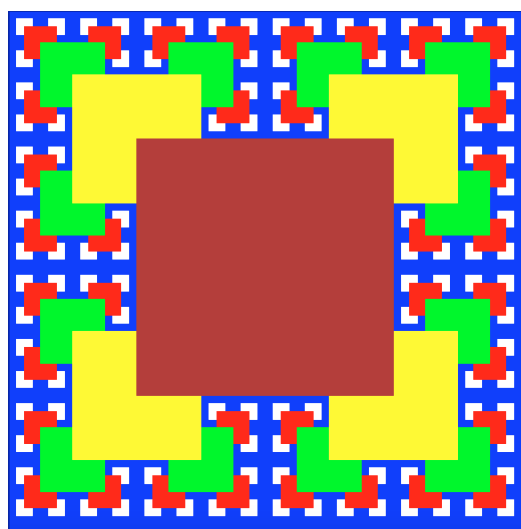
Exercice 13 (Des étoiles)

1. Ecrivez une fonction `Etoile5(taille)` traçant une étoile à cinq branches dont la longueur `taille` d'une branche est donnée.
2. Utilisez cette fonction pour reproduire le modèle ci-dessous :

**Exercice 14 (Echiquier)**

1. Ecrivez un programme permettant de tracer un échiquier dans une fenêtre de taille 800×800 .
2. Modifiez le programme précédent afin de pouvoir tracer un échiquier dont la taille est adaptée automatiquement à la taille de la fenêtre graphique dont les dimensions sont fournies par l'utilisateur.

Exercice 15 (Pyramides chinoises) A l'aide d'une fonction traçant un carré de dimensions et de couleur donnés et de boucles, reproduire les pyramides chinoises suivantes (les couleurs sont au choix). On pourra utiliser une liste pour gérer les couleurs.



Exercice 16 On donne ci-dessous trois fonctions `calcul1`, `calcul2`, et `calcul3` programmées en langage Python.

```
def calcul1(a,b):
    n = 0
    while n < 20:
        n = a + b
        a = 2*a
        b = 3*b
        n = b*n
    return n
```

```
def calcul2(a,b):
    n = 0
    while n < 20:
        n = a + b
        a = 2*a
        b = 3*b
        n = b*n
    return n
```

```
def calcul3(a,b):
    n = 0
    while n < 20:
        n = a + b
        a = 2*a
        b = 3*b
        n = b*n
    return n
```

1. Quelles sont les variables utilisées dans ces fonctions ? De quel type sont-elles a priori ?
2. Quel type de boucle est utilisé ici ?
3. Recopier et compléter le tableau de variables suivant afin de déterminer ce que renvoient respectivement les instructions `calcul1(5,1)`, `calcul2(5,1)` et `calcul3(5,1)`. On ajoutera autant de lignes que nécessaire.

	a	b	n
Avant la boucle			
Après le 1er passage dans la boucle			
Après le 2ème passage dans la boucle			
Après le ... passage dans la boucle			

4. Observer les résultats obtenus et le code des fonctions proposées et expliquer les différences.

Exercice 17 Testez la fonction ci-dessous et expliquez chaque ligne de code.

```
from turtle import *
speed("fastest")

def Spirale() :
    rayon = 1
    rayonS = 100
    while (rayon < rayonS) :
        circle(rayon, 180)
        rayon += 2
```

Exercice 18 (Rebonds) Écrivez un programme permettant de simuler, dans une fenêtre graphique de taille 400×400 , le rebond d'une balle sur les deux côtés verticaux (utilisez les fonctions `shape()` et `clear()`). On se limitera à trois allers-retours.

Exercice 19 (Evolution d'un capital 2) Une somme de 2000 € est déposée sur un compte rémunéré au taux annuel de 2%.

1. Ecrivez un script affichant le nombre d'années au bout duquel le compte atteindra 3000 €.
2. Comment modifier ce script pour qu'il puisse fonctionner avec n'importe quelle somme initiale, n'importe quel taux et n'importe quelle somme finale attendue ?

Exercice 20 (Lancers de dés) A l'aide des bibliothèques `random` et `uniform`, écrire un programme permettant de simuler le lancer d'un dé à six faces et d'afficher la face obtenue.