

Exercice 1 :

Écrivez un programme qui demande à l'utilisateur la taille du rayon d'un cercle, qui lit ce rayon et qui demande à l'utilisateur :

- s'il souhaite obtenir le périmètre ; si la réponse est positive, celle-ci sera calculée et affichée.
- s'il souhaite obtenir l'aire ; si la réponse est positive, celle-ci sera calculée et affichée.

L'utilisateur répondra avec "oui" pour oui tandis que tout autre caractère sera interprété comme non.

```
from math import *
rayon = float(input("Le rayon du cercle est de :"))
perimetre = 0
aire = 0
q1 = input("Voulez-vous obtenir le périmètre ?")
q2 = ""
if q1 == "oui":
    perimetre = 2 * rayon
    print(perimetre)
q2 = input("Voulez-vous obtenir l'aire ?")
if q2 == "oui":
    aire = pi * rayon**2
    print(aire)
```

Exercice 2 :

Écrire un programme qui affiche tous les entiers de 8 jusqu'à 23 (bornes incluses) en utilisant un for, puis qui affiche tous les entiers de 30 jusqu'à 45 (bornes incluses) en utilisant un while.

```
for i in range(8,23+1):
    print(i)
n = 30
while n <= 45:
    print(n)
    n+=1
```

Exercice 3 :

Écrire un programme qui provoque l'affichage ci-contre :

Il y a 20 lignes, et la vingtième contient 20 symboles « # »

```
#
##
###
####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

```
nb = 0
txt = "#"
for loop in range(20):
    nb += 1
    print(nb*txt)
```

Exercice 4 :

Écrire un programme qui demande à l'utilisateur de saisir des entiers positifs. Le programme s'arrête dès qu'un entier négatif est saisi. Il affiche alors le nombre d'entiers positifs qui ont été saisis.

```
nb = 0
pos = -1
while nb >= 0:
    nb = int(input("saisir un nombre positif:"))
    pos += 1
print(pos)
```

Exercice 5 :

Écrire un programme qui demande à l'utilisateur de taper un entier $N > 10$ et qui calcule la somme des cubes consécutifs de 5^3 à N^3 .

(Si possible créer une fonction cube que vous utiliserez ensuite dans le corps du programme).

```
def cube(n):
    cube = n**3
    return(cube)
nb = int(input("Choisissez un nombre supérieur à 10:"))
i = 5
somme = 0
if nb > 10:
    while i <= nb:
        print(cube(i))
        somme+=cube(i)
        i+=1
print(somme)
```

Exercice 6 :

Écrire un programme qui calcule la somme des entiers positifs inférieurs à 1000 qui ne sont ni divisibles par 3 ni divisible par 5.

```
nb = 0
resultat = 0
n = 0
o = 0
while nb <= 1000:
    n = nb%3
    o = nb%5
    if (n != 0 & o != 0):
        resultat += nb
    nb += 1
print(nb)
```

Exercice 7 (pour les plus pro) :

Écrivez un programme qui demande à l'utilisateur d'entrer une série d'entiers positifs, puis affiche le plus grand d'entre eux, ainsi que la position à laquelle il a été entré (si celui-ci a été entré plus d'une fois, c'est sa première position qui est affichée). La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro ou un entier positif impair multiple de cinq. De plus, si l'utilisateur entre un entier négatif, le programme l'ignore et poursuit son exécution, comme illustré dans l'exemple ci-dessous :

Entrez le nombre : 17 est compté n° 1

Entrez le nombre : -5 non compté

Entrez le nombre : 22 est compté n°2

Entrez le nombre : -1 non compté

Entrez le nombre : 22 non compté car déjà entré

Entrez le nombre : 15 est compté n°3

Le plus grand nombre est 22 c'est le nombre n°2

```
nb = 11
n = 0
i = 0
e = 0
while 0 != nb%10:
    nb = int(input("Entrez le nombre:"))
    i+= 1
    if nb > n & 0 != nb%10:
        n = nb
        e = i
print("Le plus grand nombre est",n, "c'est le nombre n°",e)
```

Exercice 8 : Le but de notre programme est de déterminer si une année saisie par l'utilisateur est bissextile.

Il s'agit d'un sujet très prisé des enseignants en informatique quand il s'agit d'expliquer les conditions. Je vous rappelle les règles qui déterminent si une année est bissextile ou non. Une année est dite bissextile si c'est un multiple de 4, sauf si c'est un multiple de 100.

Toutefois, elle est considérée comme bissextile si c'est un multiple de 400.

Je développe :

- Si une année n'est pas multiple de 4, on s'arrête là, elle n'est pas bissextile.
- Si elle est multiple de 4, on regarde si elle est multiple de 100.
- Si c'est le cas, on regarde si elle est multiple de 400.
- Si c'est le cas, l'année est bissextile.
- Sinon, elle n'est pas bissextile.
- Sinon, elle est bissextile.

```
years = int(input("Quelle année ?"))
reste = 0
reste = years%400
if reste == 0:
    print("L'année est bissextile")
else:
    reste = years%4
    if reste == 0:
        reste = years%100
        if reste == 0:
            print("L'année n'est pas bissextile")
        else:
            print("L'année est bissextile")
if reste != 0:
    print("L'année n'est pas bissextile")
```

Exercice 9

sans importer de bibliothèques
Faire afficher le tableau suivant

dec	b i n a i r e	hexa
0	0 0 0 0 0 0 0 0	0x 00
1	0 0 0 0 0 0 0 1	0x 01
2	0 0 0 0 0 0 1 0	0x 02
3	0 0 0 0 0 0 1 1	0x 03
4	0 0 0 0 0 1 0 0	0x 04
5	0 0 0 0 0 1 0 1	0x 05
6	0 0 0 0 0 1 1 0	0x 06
7	0 0 0 0 0 1 1 1	0x 07
8	0 0 0 0 1 0 0 0	0x 08
9	0 0 0 0 1 0 0 1	0x 09
10	0 0 0 0 1 0 1 0	0x 0A
11	0 0 0 0 1 0 1 1	0x 0B
12	0 0 0 0 1 1 0 0	0x 0C
13	0 0 0 0 1 1 0 1	0x 0D
14	0 0 0 0 1 1 1 0	0x 0E
15	0 0 0 0 1 1 1 1	0x 0F
16	0 0 0 1 0 0 0 0	0x 10
...		
250	1 1 1 1 1 0 1 0	0x FA
251	1 1 1 1 1 0 1 1	0x FB
252	1 1 1 1 1 1 0 0	0x FC
253	1 1 1 1 1 1 0 1	0x FD
254	1 1 1 1 1 1 1 0	0x FE
255	1 1 1 1 1 1 1 1	0x FF

option pour les archi pros : n'afficher que les lignes avec 3 bits à 1

```
def bin(s):
    reste = 0
    nbResultat = 0
    resultat = ""
    nbZ = 0
    while i < s:
        reste = s%2
        if resultat != "":
            resultat = format(reste) + " " + resultat
        else:
            resultat = format(reste)
        s = s // 2
        nbResultat += 1
    nbZ = 8 - nbResultat
    for loop in range(nbZ):
        if resultat != "":
            resultat = "0 " + resultat
        else:
            resultat = "0"
    return(resultat)

def nbBits(s):
    reste = 0
    nbBit = 0
    while i < s:
```

```

        reste = s%2
        if reste == 1:
            nbBit += 1
        s = s // 2
    return(nbBit)

def octale(s):
    reste = 0
    nbResultat = 0
    resultat = ""
    nbZ = 0
    while i < s:
        reste = s%8
        resultat = format(reste) + resultat
        s = s // 8
        nbResultat += 1
    nbZ = 3 - nbResultat
    for loop in range(nbZ):
        resultat = "0" + resultat
    return(resultat)

def hexad(s):
    reste = ""
    nbResultat = 0
    resultat = ""
    nbZ = 0
    while i < s:
        reste = format(s%16)
        if reste == "10":
            reste = "A"
        if reste == "11":
            reste = "B"
        if reste == "12":
            reste = "C"
        if reste == "13":
            reste = "D"
        if reste == "14":
            reste = "E"
        if reste == "15":
            reste = "F"
        resultat = format(reste) + resultat
        nbResultat += 1
        s = s // 16
    nbZ = 2 - nbResultat
    for loop in range(nbZ):
        resultat = "0" + resultat
    return(resultat)

i = 0
dec = 0
binaire = ""
hexa = ""
txt = ""
octet = ""

##Contour du tableau
for loop in range(35):

```

```

        print("-", end="")
print()
print("|dec | b i n a i r e | hexa | oct|")
print("|", end="")
for loop in range(33):
    print("-", end="")
print("|")

for loop in range(256):
    if dec <= 16 or dec >= 250:
        if nbBits(dec)== 3:
            txt = format(dec) + "|"
            if dec < 10:
                print("| ", txt, end=" ")
                binaire = bin(dec)
            elif dec < 100:
                print("| ", txt, end=" ")
                binaire = bin(dec)
            else:
                print("|", txt, end=" ")
                binaire = bin(dec)
            print(binaire, end="|")

            hexa = "0x " + hexad(dec)
            print(hexa, end=" | ")

            octet = octale(dec)
            print(octet, end="")
            print("|")
        elif dec == 17:
            print(" . . .")
        dec += 1

for loop in range(35):
    print("-", end="")

```