

```
# Calcule le pgcd et le ppcm de a et b

a = int(input('Entrer a: '))
b = int(input('Entrer b: '))

# On veut a > b
if a < b:
    aa = a
    a = b
    b = aa

# Initialisations
aa = a
bb = b
reste = aa % bb

# Boucle de calcul
while reste != 0:
    aa = bb
    bb = reste
    reste = aa % bb

# Résultats
pgcd = bb
ppcm = (a * b) // pgcd

# Affichage
print("pgcd(", a, ",", b, ") = ", pgcd)
print("ppcm(", a, ",", b, ") = ", ppcm)
```

```
import random

n = int(input('n ? '))
random.seed()
plus60piles = 0 # nombre de tirages avec plus de 60 piles sur 100

for i in range(n):
    nombrePiles = 0
    for j in range(100):
        # tirage aléatoire : 0 = pile, 1 = face
        if random.randrange(2) == 0:
            nombrePiles += 1
    print("nombre de piles ", nombrePiles)
    if nombrePiles >= 60:
        plus60piles += 1
print(plus60piles)
```

```
# Saisie d'une liste de n éléments et inversion de la liste.  
  
n = int(input("Taille de la liste : "))  
liste = []  
for i in range(n):  
    liste.append(int(input("Entrer un entier : ")))  
print(liste)  
liste.reverse()  
print(liste)
```

```
n = int(input("Entrer n : "))
l = []
for i in range(1,n//2+1):
    if n % i == 0:
        l.append(i)
l.append(n)
print(l)
```

```
import turtle

# Flèches de Xenon à 2 cibles
turtle.reset()
turtle.title("Xenon 2 points")
turtle.penup()
Ax = int(input('abscisse point A ? '))
Ay = int(input('ordonnée point A ? '))
Bx = int(input('abscisse point B ? '))
By = int(input('ordonnée point B ? '))
n = int(input('nombre de lancers ? '))

# Position de Xenon
Xx = int(input('abscisse Xenon ? '))
Xy = int(input('ordonnée Xenon ? '))
turtle.goto(Xx, Xy)
turtle.dot(10, "blue")
turtle.goto(Ax, Ay)
turtle.dot(10, "red")
turtle.goto(Bx, By)
turtle.dot(10, "green")

versA = True # la 1ère cible est A
for i in range(n):
    if versA:
        Xx = (Xx + Ax) // 2
        Xy = (Xy + Ay) // 2
    else:
        Xx = (Xx + Bx) // 2
        Xy = (Xy + By) // 2
    turtle.goto(Xx, Xy)
    turtle.dot(4, "black")
    versA = not versA
```

```
import turtle

# Flèches de Xénon avec un nombre de cibles fixées par l'utilisateur.
turtle.reset()
turtle.title("Xenon général")
turtle.penup()
nCibles = int(input('Nombre de cible ? '))
nLancers = int(input('Nombre de lancers ? '))
cibles = []
for i in range(nCibles):
    print("Abscisse de la cible ", i+1, " ? ", end = "")
    x = int(input())
    print("Ordonnée de la cible ", i+1, " ? ", end = "")
    y = int(input())
    cibles += [[x, y]]
#print(cibles)

# Position de Xenon
Xx = int(input("Abscisse de départ de Xenon ? "))
Xy = int(input("Ordonnée de départ de Xenon ? "))
turtle.goto(Xx, Xy)
turtle.dot(10, "blue")
# Positions des cibles
for i in range(nCibles):
    x = cibles[i][0]
    y = cibles[i][1]
    turtle.goto(x, y)
    turtle.dot(10, "red")
    turtle.goto(x+10, y)
    turtle.write(i+1, font=('Arial', 16))

for i in range(nLancers):
    iCible = i % nCibles
    Cx = cibles[iCible][0]
    Cy = cibles[iCible][1]
    Xx = (Xx + Cx) // 2
    Xy = (Xy + Cy) // 2
    turtle.goto(Xx, Xy)
    turtle.dot(4, "blue")
```

```
# ce programme dessine une etoile à N branches (N impair)

import turtle

print("Dessin d'une etoile impaire a N branches")
N = int(input("Entrez N : "))

# initialisations
turtle.reset()
turtle.width(2)
turtle.color("red")
turtle.speed(8)
angle = float(180 / N)
diametre = 200

# on va sur la premiere pointe
turtle.penup()
turtle.goto(0,-(diametre // 2))
turtle.pendown()
turtle.left(90) # oriente la tortue vers le Nord

# boucle de dessin
for i in range(N):
    turtle.forward(diametre)
    turtle.left(180 - angle)
```