

TP 1 : CONCEPTS DE BASE EN PROGRAMMATION PYTHON

Objectifs du TP

- Maîtriser la syntaxe de Python
- Travailler la conception algorithmique.

Exercice 1 :

Ecrire un programme qui calcule le factoriel d'un entier positif entré par l'utilisateur.

```
n = int(input("Entrez un entier positif : "))
if n < 0:
    print("Le factoriel n'est pas défini pour les entiers négatifs.")
else:
    f = 1
    for i in range(2, n + 1):
        f = f * i
    print(f"Le factoriel de {n} est : {f}")
```

Exercice 2 :

Ecrire un programme fait la résolution des équation premier degré de la forme : $ax + b = 0$

```
a = float(input("Entrez a : "))
b = float(input("Entrez b : "))
if a != 0:
    x = -b / a
    print(f"La solution de l'équation est : x = {x}")
elif b == 0:
    print("Infinité de solutions.")
else:
    print("Pas de solution.")
```

Exercice 3 :

Ecrire un programme qui vérifie si un nombre est parfait ou non.

```
n = int(input("Entrez un nombre entier positif : "))
somme = 0
for i in range(1, n):
    if n % i == 0:
        somme += i

if somme == n:
    print(f"{n} est un nombre parfait.")
else:
    print(f"{n} n'est pas un nombre parfait.")
```

Exercice 4 :

Implémentez l'algorithme d'Euclide pour trouver le plus grand commun diviseur (PGCD) de deux nombres entiers a et b.

Version 1 : Méthode Euclide

```
a = int(input("Entrez le premier nombre : "))
b = int(input("Entrez le deuxième nombre : "))

while b != 0:
    a, b = b, a % b

print(f"Le PGCD est : {a}")
```

Version 2 :

```
a = int(input("Entrez le premier nombre : "))
b = int(input("Entrez le deuxième nombre : "))

p = 1
for i in range(2, min(a,b)):
    if a%i==b%i==0:
        p = i

print(f"Le PGCD est : {p}")
```

Exercice 5 :

Ecrire un programme qui affiche le nombre d'occurrences d'un chiffre c dans un entier n entré par l'utilisateur. On suppose que n est positif.

```
n = int(input("Entrez un entier positif : "))
c = input("Entrez le chiffre à compter : ")
compteur = 0

for chiffre in str(n):
    if chiffre == c:
        compteur += 1

print(f"Le chiffre {c} apparaît {compteur} fois dans {n}.")
```

Exercice 6 :

Un nombre narcissique (ou nombre d'Armstrong de première espèce est un entier naturel n non nul qui est égal à la somme des puissances p-ièmes de ses chiffres en base dix, où p désigne le nombre de chiffres de n :

Exemples :

$$153 = 1^3 + 5^3 + 3^3$$

$$548834 = 5^6 + 4^6 + 8^6 + 8^6 + 3^6 + 4^6$$

Ecrire un programme qui vérifie si un entier n entré par l'utilisateur est narcissique ou non.

```
n = int(input("Entrez un entier positif : "))
s = str(n) # convertir le nombre en string
p = len(s)
somme = 0

for chiffre in s:
    somme += int(chiffre) ** p

if somme == n:
    print(f"{n} est un nombre narcissique.")
else:
    print(f"{n} n'est pas un nombre narcissique.")
```

Exercice 7 :

Écrire un programme qui calcule le n-ième terme de la suite de Fibonacci, défini par :

$$\begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-2} + U_{n-1} \end{cases}$$

```
n = int(input("Entrez un entier positif n : "))
if n == 0:
    print("U0 = 0")
elif n == 1:
    print("U1 = 1")
else:
    u0, u1 = 0, 1
    for i in range(2, n + 1):
        u0, u1 = u1, u0 + u1
    print(f"Un = {u1}")
```

Exercice 8 :

Un joueur lance 10 fléchettes. Chaque fléchette rapporte 0, 20 ou 50 points selon la zone touchée. Il a obtenu 300 points. Ecrire un programme pour trouver les différentes possibilités.



```
total_points = 300
max_lance = 10
for a in range(max_lance + 1): # 0 à 10 fléchettes à 0 points
    for b in range(max_lance + 1 - a): # 0 à (10 - a) fléchettes à 20 points
        c = max_lance - a - b # Fléchettes restantes à 50 points
        points = a * 0 + b * 20 + c * 50
        if points == total_points:
            print(f"Fléchettes 0: {a}, Fléchettes 20: {b}, Fléchettes 50: {c}")
```