



STRUCTURE DE DONNEES

Exercice 1

Créer un programme qui permet d'afficher tous les nombres pairs compris entre 2 et un nombre entier saisi par l'utilisateur, en utilisant deux méthodes différentes : une boucle "for" et une boucle "while".

```
n = int(input("Entrez un nombre : "))
for i in range(2, n+1, 2):
    print(i)

# avec la boucle while on a :
n = int(input("Entrez un nombre : "))
i = 2
while i <= n:
    print(i)
    i += 2
```

1

Exercice 2

Créer un programme qui permet d'afficher la table de multiplication d'un nombre saisi par l'utilisateur.

```
n = input("Entrez un nombre : ")
for i in range(1, 11):
    print(n, "x", i, "=", n*i)
```

Exercice 3

Créer un programme qui permet de compter le nombre de chiffres dans un entier :

```
n = int(input("Entrez un nombre : "))
count = 0
while n != 0:
    n //= 10
    count += 1
print("Nombre de chiffres : ", count)
```

Exercice 4

Créer un programme qui permet d'afficher le premier et le dernier chiffre d'un nombre entré par l'utilisateur

```
n = int(input("Entrez un nombre : "))
last_digit = n % 10
while n >= 10:
    n //= 10
first_digit = n
print("Premier chiffre :", first_digit)
print("Dernier chiffre :", last_digit)
```



Exercice 5

Écrire un programme qui demande à l'utilisateur d'entrer un nombre, puis renverse la place du premier le dernier chiffre de ce nombre.

```
n = input("Entrez un nombre : ")
print("Nouveau nombre :", n[-1] + n[1:-1] + n[0])
```

Exercice 6

Écrire un programme qui demande à l'utilisateur d'entrer un nombre, puis calcule la somme de tous ses chiffres.

```
n = input("entrer un nombre : ")
print(sum([int(i) for i in n]))
```

Exercice 7

Écrire un programme qui demande à l'utilisateur d'entrer un nombre, puis détermine si ce nombre est un palindrome.

```
def IsPalindrom(n):
    return n == n[::-1]

nombre = input("Entrez un nombre : ")
if IsPalindrom(nombre):
    print("Ce nombre est un palindrome")
else:
    print("Ce nombre n'est pas un palindrome")
```

Exercice 8

Écrire un programme qui demande à l'utilisateur d'entrer un nombre, puis détermine la fréquence de tous les nombres qui le compose

```
n = int(input("Entrez un nombre : "))
digits_count = [0]*10
while n != 0:
    digit = n % 10
    digits_count[digit] += 1
    n //= 10
for i in range(10):
    if digits_count[i] > 0:
        print("Le chiffre", i, "apparaît", digits_count[i], "fois.")
```



Exercice 9

Écrire un programme qui demande à l'utilisateur d'entrer un nombre entier, puis affiche le mot correspondant à chaque chiffre de ce nombre.

```
num = input("Entrez un nombre : ")
mots = ["zéro", "un", "deux", "trois", "quatre", "cinq", "six", "sept", "huit", "neuf"]
for chiffre in num:
    print(mots[int(chiffre)])
```

3

Exercice 10

Écrire un programme qui demande à l'utilisateur d'entrer un nombre entier, puis qui calcule et affiche tous les facteurs de ce nombre.

```
nombre = int(input("Entrez un nombre : "))
facteurs = []
for i in range(1, nombre+1):
    if nombre % i == 0:
        facteurs.append(i)
print("Les facteurs de", nombre, "sont :", facteurs)
```

Exercice 11

Écrire un programme qui demande à l'utilisateur d'entrer un nombre entier, puis qui calcule et affiche la factorielle de ce nombre.

```
from math import factorial as f
n = int(input("Entrez un nombre : "))
print(f(n))

#OU

nombre = int(input("Entrez un nombre : "))
factorielle = 1
for i in range(1, nombre+1):
    factorielle *= i
print("La factorielle de", nombre, "est :", factorielle)
```

Exercice 12

Écrire un programme qui affiche les nombres de 0 à 99 à mais qui s'arrête à 8 en utilisant break.

```
for i in range(100):
    if i == 8:
        break
    print(i)
```



Exercice 13

Ecrire un programme qui affiche tous les éléments de la liste

weather = ["snow", "rain", "sun", "clouds"] sauf 'sun'

```
weather = ["snow", "rain", "sun", "clouds"]

for iterator in weather:
    if iterator == "sun":
        continue
    print(iterator)
```

*« Quelle prétention de prétendre que
l'informatique est récente: Adam et Eve avaient déjà un Apple! »*