

---

# IngeSUP - TD 03 - Les structures de répétition

---

*"Le monde est une machine ingénieuse de création et de destruction en boucle sans fin."*

Casharel Nth

---

## Exercice 03.1

---

1. Ecrivez un programme qui demande un nombre (entier ou réel) à rentrer au clavier tant que celui-ci n'est pas négatif.

Entrée [6]:



```
x=float(input("entrez un nombre négatif"))
while x>0:
    print("Ce n'est pas un nombre négatif !!")
    x=float(input("entrez un nombre négatif"))
```

```
entrez un nombre négatif6
Ce n'est pas un nombre négatif !!
entrez un nombre négatif7
Ce n'est pas un nombre négatif !!
entrez un nombre négatif-2
```

2. Ecrivez un programme qui demande de saisir deux nombre entiers tant que la somme de ces deux nombres n'est pas un nombre pair.

Entrée [7]:



```
a=float(input("entrez un nombre A"))
b=float(input("entrez un nombre B"))
while (a+b)%2!=0:
    print("la somme n'est pas paire")
    a=float(input("entrez un nombre A"))
    b=float(input("entrez un nombre B"))
print("ok, la somme fait", a+b)
```

```
entrez un nombre A3
entrez un nombre B4
la somme n'est pas paire
entrez un nombre A1
entrez un nombre B2
la somme n'est pas paire
entrez un nombre A1
entrez un nombre B7
ok, la somme fait 8.0
```

---

## Exercice 03.2

---

1. Traduisez en python l'algorithme suivant

```
DEMANDER a et b
TANTQUE b <= a
    a = a - b
FIN TANTQUE
AFFICHER a
```

Entrée [8]:



```
a=float(input("entrez un nombre A"))
b=float(input("entrez un nombre B"))
while b<=a:
    a=a-b
print(a)
```

```
entrez un nombre A14
entrez un nombre B3
2.0
```

2. Qu'affiche le programme si on entre 14 pour a et 3 pour b ?

**Réponse:**

3. Plus généralement, si a et b sont deux entiers strictement positifs. Que fait ce programme ?

Réponse:

---

## Exercice 03.3

---

Ecrivez un programme qui affiche une variable `somme_entiers` correspondant à la somme des entiers de 3 à 172, c'est à dire  $3 + 4 + 5 + \dots + 171 + 172$ .

Entrée [11]:



```
somme=0
for i in range(3,173):
    somme=somme+i

print(somme)
```

14875

---

## Exercice 03.4

---

Ecrire un programme qui calcule et affiche le plus grand entier  $n$  tel que:

$$(n + 1)(n + 2) < 2020$$

Entrée [13]:



```
i=0
while (i+1)*(i+2)<2020:
    i=i+1
print(i-1)
```

43

---

## Exercice 03.5

---

Nous savons qu'un capital  $C_0$  placé à 5 % rapporte au bout d'un an  $C_0 \times 0,05$  d'intérêts.

Ces intérêts ajoutés au capital nous donnent un nouveau capital  $C_1 = 1,05 \times C_0$ .

En recommençant le processus chaque année, on crée ainsi une suite géométrique de raison 1,05 car :  
 $C_{k+1} = 1,05 \times C_k$ .

Réalisez un programme permettant de calculer le capital obtenu au bout d'un nombre d'années  $N$  avec un Capital initial  $C_0$  et un taux d'intérêt  $\text{taux}$ . Les variables  $N$ ,  $C_0$  et  $\text{taux}$  doivent être saisies par l'utilisateur. Le programme affichera le capital  $C$  correspondant à chaque année.

## Test

Testez avec les paramètres  $N=3$ ,  $C_0=20000$  et  $\text{taux}=2$ .

Entrée [15]:



```
N=int(input("nombre d'années ?"))
C0=float(input("capital initial ?"))
taux=float(input("taux ?"))

C=C0
for i in range(N):
    C=C*(1+taux/100)
    print(C)
```

```
nombre d'années ?10
capital initial ?20000
taux ?5
21000.0
22050.0
23152.5
24310.125
25525.631250000002
26801.912812500002
28142.008453125003
29549.108875781254
31026.56431957032
32577.892535548835
```

---

## Exercice 03.6

---

On considère la suite définie par récurrence de la façon suivante:

$$x_{k+1} = \frac{1}{2} \left( x_k + \frac{a}{x_k} \right)$$

Réalisez un programme permettant de calculer l'élément  $x$  tandis que les variables  $a$ ,  $x_0$  et  $k$  seront renseignées au clavier par l'utilisateur. Afficher  $x$ .

Entrée [1]:



```
a=float(input("entrez a"))
x0=float(input("entrez x0"))
k=int(input("entrez k un entier"))

x=x0

for i in range(k):
    x=(1/2)*(x+a/x)

    print(x)
```

```
entrez a1
entrez x02
entrez k un entier3
1.25
1.025
1.0003048780487804
```

---

## Exercice 03.7

---

Ecrivez un programme qui calcule le nombre de lettres qui composent une variable `mot` que vous définirez. Le résultat sera stocké derrière la variable `nbre_ltr` .

N'oubliez pas la boucle `for..in`

Entrée [19]:



```
mot=input("entrez un mot")

compteur=0

for lettre in mot:
    compteur += 1

print(compteur)

#on peut aussi faire
print(len(mot))
#mais l'objectif est de s'entrainer sur les boucles
```

```
entrez un motbonjour
7
7
```

---

## Exercice 03.8

Une séquence d'ADN valide signifie qu'elle n'est pas vide et est formée exclusivement d'une combinaison arbitraire de "A", "C", "G" ou "T" (A pour Adénine, C pour Cytosine, G pour Guanine et T pour Thymine).

Proposez un programme qui, après avoir demandé à l'utilisateur de saisir au clavier une chaîne d'ADN `adn`, stocke derrière la variable `res` la chaîne de caractère `ADN valide` si la chaîne saisie est bien une chaîne d'ADN. Le programme stockera `ADN non valide` le cas échéant.

Entrée [21]:

```
adn=input("entrez un adn (composé de A-C-G-T)")
res="adn valide"

for lettre in adn:
    if lettre!="A" and lettre != "C" and lettre != "G" and lettre != "T":
        res="adn non valide"
        break # si on trouve une lettre problématique, pas la peine de vérifier la suite or

print(res)
```

```
entrez un adn (composé de A-C-G-T)ACTGTCFGCT
adn non valide
```

## Exercice 03.9

1. Ecrire un programme python qui demande à l'utilisateur de saisir un entier `n` compris entre 2 et 12. Si l'utilisateur se trompe et saisit un nombre qui ne vérifie pas la condition le programme doit lui redemander la saisie jusqu'à ce qu'il saisisse un entier `n` vérifiant la condition.

Entrée [ ]:

```
n=int(input("entrez un entier entre 2 et 12"))

while n<2 or n>12:
    print("erreur")
    n=int(input("entrez un entier entre 2 et 12"))
```

2. Compléter le programme précédent pour qu'il affiche toutes les combinaisons possibles pour faire le nombre `n` en lançant deux dés à six faces.

Entrée [23]:



```
n=int(input("entrez un entier entre 2 et 12"))

while n<2 or n>12:
    print("erreur")
    n=int(input("entrez un entier entre 2 et 12"))

for i in range(1,7):
    for j in range(1,7):
        if i+j==n:
            print(i,j)
```

entrez un entier entre 2 et 128

2 6  
3 5  
4 4  
5 3  
6 2

---

## Exercice 03.10

---

1. En vous inspirant de l'exemple du cours et à l'aide d'une boucle for, affichez la table de 8

Entrée [24]:



```
for i in range(1,11):
    print(i, "* 8 = ", i*8)
```

1 \* 8 = 8  
2 \* 8 = 16  
3 \* 8 = 24  
4 \* 8 = 32  
5 \* 8 = 40  
6 \* 8 = 48  
7 \* 8 = 56  
8 \* 8 = 64  
9 \* 8 = 72  
10 \* 8 = 80

2. A l'aide de **deux boucle for imbriquées** et en vous inspirant de la question précédente, affichez toutes les tables de multiplication de 1 à 10.

Entrée [27]:



```
for n in range(1,11):  
    print("table de", n)  
    for i in range(1,11):  
        print(i, "*",n,"=", i*n)
```

table de 1

1 \* 1 = 1

2 \* 1 = 2

3 \* 1 = 3

4 \* 1 = 4

5 \* 1 = 5

6 \* 1 = 6

7 \* 1 = 7

8 \* 1 = 8

9 \* 1 = 9

10 \* 1 = 10

table de 2

1 \* 2 = 2

2 \* 2 = 4

3 \* 2 = 6

4 \* 2 = 8

5 \* 2 = 10

6 \* 2 = 12

7 \* 2 = 14

