

---

# IngeSUP - TD 03 - Les structures de répétition

---

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

CashareI 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 [ ]:



```
# VOTRE CODE ICI
```

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 [ ]:



```
# VOTRE CODE ICI
```

---

## 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 [ ]:



# VOTRE CODE ICI

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 [ ]:



# VOTRE CODE ICI

## Exercice 03.4

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

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

Entrée [ ]:



# VOTRE CODE ICI

## 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 [ ]:



# VOTRE CODE ICI

---

## 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 [ ]:



# VOTRE CODE ICI

---

## 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 [ ]:



# VOTRE CODE ICI

## 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 [ ]:



# VOTRE CODE ICI

## 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 [ ]:



# VOTRE CODE ICI

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 [ ]:



# VOTRE CODE ICI

---

## 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 [ ]:



```
# VOTRE CODE ICI
```

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 [ ]:



```
# VOTRE CODE ICI
```

---

## Corrigé du TD 03

Vous pouvez retrouver le corrigé de ce TD [ici \(Corrig%C3%A9s/Corrig%C3%A9\\_TD%2003.ipynb\)](#).