
Algorithmique et Programmation 1 – TD - TP 3

BOUCLES

CORRECTION

Exercice 1 - Premières boucles **while**

Indiquez ce que les programmes suivants affichent :

1. Suite d'instruction 1 :

```
i = 1
while i <= 5 :
    print(i)
    i = i+1
```

```
1
2
3
4
5
```

2. Suite d'instructions 2 :

```
i = 1
while i <= 5 :
    i = i+1
    print(i)
```

```
2
3
4
5
6
```

3. Suite d'instructions 3 :

```
i = 1
while i > 0 :
    print(i)
    i = i + 1
```

```
# Boucle infinie
```

4. Suite d'instructions 4 :

```
i = 1
while i > 10 :
    print(i)
    i = i + 1
```

```
# Ne rentre pas dans la boucle
```

Exercice 2 - Premières boucles **for**

Indiquez ce que les programmes suivants affichent :

1. Suite d'instruction 1 :

```
for i in range(50, 100, 10) :  
    print(i)
```

```
50  
60  
70  
80  
90
```

2. Suite d'instructions 2 :

```
for i in range(50, 100, 10) :  
    print(10)
```

```
10  
10  
10  
10  
10
```

3. Suite d'instructions 3 :

```
x = 0  
for i in range(0, 5, 1) :  
    x = x + 5  
print(x)
```

```
25
```

4. Suite d'instructions 4 :

```
x = 0  
for i in range(5) :  
    x = x + 5  
    print(x)
```

```
5  
10  
15  
20  
25
```

Exercice 3 - Table de simulation

Ecrire les tables de simulation des deux programmes suivants, et en déduire ce qui est affiché en sortie.

1. Programme 1 :

```
n = 6
i = 1
compte = 0

while i <= n :
    if n % i == 0 :
        compte = compte + 1
    i = i + 1

print(compte)
```

tour de boucle	var. i	var. compte	condition while	condition if
entrée	1	0	$1 \leq 6$: True	$6\%1 == 0$: True
tour 1	2	1	$2 \leq 6$: True	$6\%2 == 0$: True
tour 2	3	2	$3 \leq 6$: True	$6\%3 == 0$: True
tour 3	4	3	$4 \leq 6$: True	$6\%4 == 0$: False
tour 4	5	3	$5 \leq 6$: True	$6\%5 == 0$: False
tour 5	6	3	$6 \leq 6$: False	$6\%6 == 0$: True
tour 6	7	4	$7 \leq 6$: False	Non applicable

Retourne le nombre de diviseurs de n

2. Programme 2 :

```
n = int(input("Donnez un entier : "))

compte = 0

while n % 2 == 0 :
    compte = compte + 1
    n = n / 2

print(compte)
```

Pour pouvoir écrire la table de simulation, il faut d'abord fixer un n . Par exemple, $n = 12$

tour de boucle	variable n	variable compte	condition
entrée	12	0	$12\%2 == 0$: True
tour 1	6	1	$6\%2 == 0$: True
tour 2	3	2	$3\%2 == 0$: False

La variable `compte` retourne le nombre de fois où un nombre est divisible par 2.

Exercice 4 - Comprendre et modifier un programme

Soit le programme suivant :

```
s = 0

for i in range(0, 101, 2) :
    s = s + i * i

print(s)
```

1. Expliquez ce que fait ce programme

Ce programme calcule la somme des carrés des entiers pairs de 0 à 100

2. Modifiez ce programme de façon à ce qu'il calcule la somme des cubes de 0 à 42

```
s = 0

for i in range(0, 43, 1) :
    s = s + (i * i * i)

print(s)
```

Élément de cours : utiliser sep et end dans la fonction print

Il est possible de modifier le **séparateur** et la **fin de ligne** dans la commande print. Quelques exemples :

- `print(3, 5)` affiche 3 5
- `print(3, 5, sep="+++")` affiche 3+++5
- `print(3, 5, sep="-", end="@")` affiche 3-5@ sans retour à la ligne ensuite

Exercice 5 - Comprendre et modifier un programme (2)

Soit le programme suivant :

```
nombre_notes = 5

print("Veuillez entrer la première note")
note_saisie=float(input())

while note_saisie > 20 or note_saisie < 0 :
    print("Attention! une note est comprise entre 0 et 20")
    print("Veuillez entrer la première note")
    note_saisie=float(input())

note_min = note_saisie
note_max = note_saisie
indice_min = 1
indice_max = 1

for i in range(2, nombre_notes + 1) :
    print("Veuillez entrer la note numéro", i)
    note_saisie=float(input())

    while note_saisie > 20 or note_saisie < 0 :
        print("Attention! une note est comprise entre 0 et 20")
        print("Veuillez entrer la note numéro", i)
        note_saisie=float(input())

    if note_min > note_saisie :
        note_min = note_saisie
        indice_min = i
    if note_max < note_saisie :
        note_max = note_saisie
        indice_max = i

print(nombre_notes, note_min, indice_min, note_max, indice_max)
```

1. Expliquez ce que fait ce programme

Ce programme demande 5 notes à l'utilisateur, vérifie que ces notes sont comprises entre 0 et 20 (et sinon, redemande la note autant de fois que nécessaire), et calcule, puis affiche, la note min et la note max, et les premières positions dans lesquelles elles ont été rentrées.

2. Recopiez-le dans un fichier `notes.py`, et vérifiez que votre réponse précédente est correcte

3. Modifiez le print final afin que l'affichage des résultats soient plus clairs.

```
nombre_notes = 5

print("Veuillez entrer la première note")
note_saisie=float(input())

while note_saisie > 20 or note_saisie < 0 :
    print("Attention! une note est comprise entre 0 et 20")
    print("Veuillez entrer la première note")
    note_saisie=float(input())

note_min = note_saisie
note_max = note_saisie
indice_min = 1
indice_max = 1

for i in range(2, nombre_notes + 1):
    print("Veuillez entrer la note numéro",i)
    note_saisie=float(input())

    while note_saisie > 20 or note_saisie < 0:
        print("Attention! une note est comprise entre 0 et 20")
        print("Veuillez entrer la note numéro",i)
        note_saisie=float(input())

    if note_min > note_saisie :
        note_min = note_saisie
        indice_min = i
    if note_max < note_saisie :
        note_max = note_saisie
        indice_max = i

print(nombre_notes, "notes ont été données.")
print("La note minimale, tapée en position " + str(indice_min) + ", est", note_min)
print("la note maximale tapée en position" + str(indice_max) + ", est", note_max)
```

4. Ajoutez le calcul de la moyenne des notes

```
nombre_notes = 5

print("Veuillez entrer la première note")
note_saisie=float(input())

while note_saisie > 20 or note_saisie < 0 :
    print("Attention! une note est comprise entre 0 et 20")
    print("Veuillez entrer la première note")
    note_saisie=float(input())

note_min = note_saisie
note_max = note_saisie
indice_min = 1
indice_max = 1
sommes_notes = note_saisie

for i in range(2, nombre_notes + 1) :
    print("Veuillez entrer la note numéro", i)
    note_saisie=float(input())

    while note_saisie > 20 or note_saisie < 0 :
        print("Attention! une note est comprise entre 0 et 20")
        print("Veuillez entrer la note numéro", i)
        note_saisie=float(input())

    somme_notes = somme_notes + note_saisie

    if note_min > note_saisie:
        note_min = note_saisie
        indice_min = i
    if note_max < note_saisie:
        note_max = note_saisie
        indice_max = i

moyenne_notes = somme_notes/nombre_notes

print(nombre_notes, "notes ont été données.")
print("La note minimale, tapée en position " + str(indice_min) + ", est", note_min)
print("la note maximale, tapée en position " + str(indice_max) + ", est", note_max)
print("La moyenne est", moyenne_notes)
```

5. Modifiez à présent ce programme de façon à ce qu'on ne connaisse pas le nombre de notes à saisir par l'utilisateur, mais que la saisie s'interrompe dès que l'utilisateur tape une note négative ou une note strictement supérieure à 20.

```
print("Veuillez entrer la première note")
note_saisie=float(input())

if note_saisie > 20 or note_saisie < 0 :
    print("Aucune note n'a été donnée")
else :
    somme_notes = note_saisie
    note_min = note_saisie
    note_max = note_saisie
    indice_min = 1
    indice_max = 1
    nombre_notes = 1
    i = 2
    arret = False

    while not(arret) :
        print("Veuillez entrer la note numéro", i)
        note_saisie=float(input())

        if note_saisie > 20 or note_saisie < 0 :
            arret = True
        else:
            somme_notes = somme_notes + note_saisie
            nombre_notes = nombre_notes + 1
            if note_min > note_saisie:
                note_min = note_saisie
                indice_min = i
            if note_max < note_saisie:
                note_max = note_saisie
                indice_max = i
            i = i + 1

moyenne_notes = somme_notes/nombre_notes
print(nombre_notes, "notes ont été saisies.")
print("La note minimale, tapée en position " + str(indice_min) + ", est", note_min)
print("la note maximale, tapée en position " + str(indice_max) + ", est", note_max)
print("La moyenne est", moyenne_notes)
```


Exercice 6 - Comprendre et modifier un programme (3)

Soit le programme suivant :

```
hauteur = int(input("Entrez un entier : "))

for i in range(1, hauteur + 1) :
    print("*" * i)
```

1. Expliquez ce que fait ce programme

```
Entrez un entier : 6
*
**
***
****
*****
*****
```

2. Recopiez-le dans un fichier `triangle.py`, et vérifiez que votre réponse précédente est correcte
3. Modifiez ce programme de façon à afficher un triangle isocèle. Exemple de déroulé :

```
Entrez un entier : 6
      *
     **
    ***
   ****
  *****
 *****
*****
```

```
hauteur = int(input("Entrez un entier : "))
nb_etoiles = 1
nb_espaces = hauteur - 1

for i in range(1, hauteur + 1) :
    print(" " * nb_espaces + "*" * nb_etoiles)
    nb_etoiles = nb_etoiles + 2
    nb_espaces = nb_espaces - 1
```

Exercice 7 - Somme

Ecrire un programme qui fait la somme des entiers tapés au clavier par l'utilisateur, et ce tant qu'il ne tape pas 0. Exemple de déroulé :

```
Donnez le premier entier : 23
Donnez l'entier suivant : 5
Donnez l'entier suivant : -6
Donnez l'entier suivant : 2
Donnez l'entier suivant : 0
La somme des entiers saisis est 24
```

```
entier = int(input("Donnez le premier entier : "))
somme = entier

while entier != 0 :
    entier = int(input("Donnez l'entier suivant : "))
    somme = somme + entier

print("La somme des entiers saisis est", somme)
```

Exercice 8 - Le chiffre 42

Ecrire un programme qui lit 5 entiers tapés au clavier par l'utilisateur, puis affiche "Gagné" si le chiffre 42 se trouvait parmi ceux-là, "Perdu" sinon. Attention, il faut lire les 5 entiers, puis ensuite seulement afficher le résultat. Exemple de déroulé :

```
Veillez entrer l'entier numéro 1 : 65
Veillez entrer l'entier numéro 2 : 42
Veillez entrer l'entier numéro 3 : 32
Veillez entrer l'entier numéro 4 : 8
Veillez entrer l'entier numéro 5 : 74
Gagné
```

```
gagne = False

for i in range(1, 6) :
    print("Veillez entrer l'entier numéro", i, end = " : ")
    chiffre = int(input())
    if chiffre == 42 :
        gagne = True

if gagne :
    print("Gagné")
else :
    print("Perdu")
```

Modifiez ensuite votre programme pour qu'il affiche "Gagné" dès que le chiffre 42 est entré par l'utilisateur, "Perdu" si 5 chiffres différents de 42 ont été saisis. Exemple de déroulé :

```
Veillez entrer l'entier numéro 1 : 65
Veillez entrer l'entier numéro 2 : 42
Gagné
```

```
gagne = False
i = 1

while not(gagne) and i <= 5 :
    print("Veillez entrer l'entier numéro", i, end = " : ")
    chiffre = int(input())
    if chiffre == 42 :
        gagne = True
    i = i+1

if gagne :
    print("Gagné")
else :
    print("Perdu")
```

Exercice 9 - Suite

Ecrire un programme qui demande à l'utilisateur un entier u_0 et un entier $term$ et qui affiche les $term$ premiers termes de la suite définie comme suit :

$$\begin{cases} u_{n+1} = 3u_n + 1 & \text{si } u_n \text{ est impair} \\ u_{n+1} = u_n/2 & \text{si } u_n \text{ est pair} \end{cases}$$

Voici un déroulé du programme attendu :

```
Donnez le premier terme u_0 de la suite : 6
Donnez le nombre de termes de la suite que vous souhaitez calculer: 4
Le terme 1 de la suite est égal à 3.0
Le terme 2 de la suite est égal à 10.0
Le terme 3 de la suite est égal à 5.0
Le terme 4 de la suite est égal à 16.0
```

```
u0 = int(input("Donnez le premier terme u_0 de la suite : "))
term = int(input("Donnez le nombre de termes de la suite que vous souhaitez calculer: "))

un = u0

for i in range(1, term+1) :
    if un % 2 == 0 :
        un1 = un/2
    else :
        un1 = 3*un + 1
    print("Le terme", i, "de la suite est égal à", un1)
    un = un1
```

Exercice 10 - Le blé du joueur d'échec

On raconte qu'il y a environ 14 siècles, un Prince de Perse voulait récompenser l'inventeur du jeu des échecs. Ce dernier lui demanda le cadeau suivant : 1 grain de blé sur la première case de l'échiquier, 2 sur la deuxième, 4 sur la troisième et ainsi de suite en doublant jusqu'à la 64ème case.

Sachant qu'un grain de blé pèse en moyenne un quart de gramme, combien de temps aurait-il fallu pour satisfaire le vœu de l'inventeur au rythme actuel de la production mondiale de blé qui est d'environ 758 millions de tonnes par an (en 2018) ?

```
nombre_case = 1 #nombre de grains de blé sur chacune des 64 cases
nombre_ble = 1 #nombre de grains de blé au total

for i in range(2, 65) :
    nombre_case = nombre_case*2
    nombre_ble = nombre_ble + nombre_case

poids_gramme = nombre_ble * 0.25 #Un grain de blé pèse 0.25 grammes
poids_tonne = poids_gramme/1000000 #Une tonne correspond à 1 000 000 grammes

annees = poids_tonne/758000000 #758 000 000 de tonnes produites par an

print(annees, "années")

# 6084.01849396753 années
```