

③

TP3 : la boucle for

Exercice 1 :

Ecrivez un programme qui demande à l'utilisateur d'entrer un entier naturel n , puis affiche les carrés des entiers de 1 à n .

Exercice 2 :

Une suite (u_n) est dite arithmético-géométrique s'il existe deux réels a et b tels que pour tout $n \in \mathbb{N}$, $u_{n+1} = au_n + b$.

1. Complétez puis tapez le programme suivant afin qu'il :

- demande à l'utilisateur de saisir deux réels a et b et la valeur de u_0 ;
- demande à l'utilisateur de saisir un entier naturel n ;
- calcule puis affiche la valeur u_n ;

```
a = plus l'input l'ajoute me 2 un réel  
b = .....  
u = .....  
n = .....  
for i in range(.....) :  
    u = .....  
print(u)
```

Ecrit
chez
moi.

Lorsqu'on saisit $a = 2$, $b = 1$, $u_0 = 4$ et $n = 15$, on obtient $u_{15} = 163839$.

Lorsqu'on saisit $a = 0.3$, $b = -5$, $u_0 = 2$ et $n = 6$, on obtient $u_6 = -7,136192$.

2. Complétez le programme précédent afin qu'il affiche la somme des termes de u_0 à u_n .

Lorsqu'on saisit $a = 2$, $b = 1$, $u_0 = 4$ et $n = 15$, on obtient $S = 327659$.

Lorsqu'on saisit $a = 0.3$, $b = -5$, $u_0 = 2$ et $n = 6$, on obtient $S = -36,941632$.

Exercice 3 :

Une légende de l'Inde ancienne raconte que le jeu d'échecs a été inventé par un vieux sage, que son roi voulut remercier en lui affirmant qu'il lui accorderait n'importe quel cadeau en récompense. Le vieux sage demanda qu'on lui fournisse simplement un peu de riz pour ses vieux jours, et plus précisément le nombre de grains de riz suffisant pour qu'on puisse déposer :

- 1 grain sur la première case
- 2 grains sur la deuxième case
- 4 grains sur la troisième case
- 8 grains sur la quatrième case ...
- D'une manière générale, 2^{i-1} grains sur la i -ème case.

Et ainsi de suite jusqu'à la 64^{ème} case.

1. Ecrivez un programme qui affiche le nombre exact de grains à déposer sur chaque case du jeu. L'affichage devra être du type :

```
1 grain sur la case 1  
2 grains sur la case 2  
4 grains sur la case 3  
8 grains sur la case 4
```

← vous pouvez pour simplifier laisser une faute d'orthographe (grains au lieu de grain) sur la première ligne

2. Modifiez-le pour faire calculer en plus le nombre total de grains de riz.

Exercice 4 :

Ecrivez un programme qui affiche la table de multiplication de 7 de 7×0 à 7×10 .

L'affichage devra être :

$7 \times 0 = 0$

$7 \times 1 = 7$

$7 \times 2 = 14$

...

Exercice 5 :

Ecrivez un programme qui affiche toutes les tables de multiplication de 1 jusqu'à 10. Vous devrez obtenir l'affichage suivant, sachant que pour faire afficher une tabulation, il faut écrire `print (... , end="\t")`, les ... étant remplacés par ce qui doit être affiché avant la tabulation. D'autre part, pour passer à la ligne suivante, il suffit de taper `print ("")`.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Exercice 6 :

Ecrivez un programme qui affiche tous les diviseurs positifs d'un nombre entier strictement positif saisi au clavier.

Indication : k est un diviseur de n si le reste de la division euclidienne de n par k vaut 0 (utiliser l'opérateur %)

Exercice 7 :

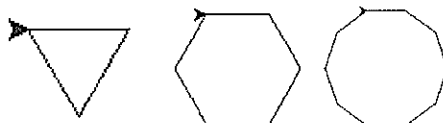
Ecrivez un programme qui demande à l'utilisateur de saisir un entier positif n et affiche n lignes de n étoiles. Par exemple, pour $n=5$, il devra afficher :

```
*****
*****
*****
*****
*****
```

Exercice 8 :

A l'aide du module turtle (voir TP2), écrivez un programme qui demande à l'utilisateur de saisir un entier positif n , puis trace un polygone régulier de côté n .

Un polygone régulier est un polygone dont tous les côtés et tous les angles sont égaux ; les angles d'un polygone régulier à n côtés valent $\frac{360}{n}$ degrés.



Exercice 9 :

A l'aide du module turtle (voir TP2), écrivez un programme qui permette de tracer une spirale :



Pour cela, il suffit de tracer un demi-cercle de rayon choisi, puis de tracer ensuite un demi-cercle de rayon plus grand, puis de tracer ensuite un demi-cercle de rayon encore plus grand...

Vous pourrez utiliser la fonction `circle(rayon, angle)` qui permet de tracer un arc de cercle de rayon donné correspondant à l'angle donné. Par exemple, l'instruction `circle(10, 90)` permet de tracer un quart de cercle de rayon 10.