

Travaux dirigés n° 2

Boucles

Utilisation de la boucle `while`

La boucle `while` sert à itérer (répéter) un bloc d'instructions tant qu'une condition est vérifiée. La condition est une expression booléenne, c'est-à-dire une expression qui est vraie (`True`) ou fausse (`False`).

*Exercice 2.1

Copiez et exécutez le programme ci-dessous. Indiquez le traitement fait par ce programme. Qu'est ce qui se passe ?

```
i=1
while i<=10:
    print("3 *",i,"=",3*i)
    i=i+1
```

*Exercice 2.2

Écrivez un programme qui demande deux entiers `a` et `b` et qui affiche les nombres `i` compris entre `a` et `b` (inclus) tel que `i` n'est pas multiple de 3 et $i^2 - 4$ est multiple de 7.

*Exercice 2.3

Écrivez un programme qui demande la saisie d'un nombre entier jusqu'à ce que le nombre saisi soit compris entre 1 et 4 (inclus).

*Exercice 2.4

Écrivez un programme qui demande en boucle à l'utilisateur d'entrer des entiers et affiche chaque entier avec son numéro d'ordre de saisie. Le programme doit s'arrêter en affichant **Merci** lorsque l'utilisateur a rentré 15 entiers ou bien s'il a saisi la valeur 0.

Exemple d'exécution :

```
Donnez un entier : 56
Entiers numéro 1 : 56
Donnez un entier : 23
Entiers numéro 2 : 23
Donnez un entier : 8
Entiers numéro 3 : 8
Donnez un entier : 0
Merci
```

*Exercice 2.5

Écrivez un programme qui demande à un joueur de trouver un nombre qui a été choisit au hasard entre 1 et 100 par la machine.

Le programme doit ensuite demander en boucle au joueur de proposer un nombre et indiquer après chaque proposition si le nombre proposé est trop petit ou trop grand.

Le programme s'arrête lorsque le joueur a trouvé le nombre et indique alors combien de coups ont été nécessaires.

Pour choisir un nombre au hasard entre 1 et 100 vous utiliserez l'appel à la fonction `randint(1,100)` qui retourne un entier compris entre 1 et 100. Cette fonction nécessite d'importer le module `random` donc d'ajouter l'instruction suivante au début de votre programme : `from random import *`

Exemple d'exécution :

Proposer un nombre entre 1 et 100 : 50
Trop petit
Proposer un nombre entre 1 et 100 : 75
Trop grand
Proposer un nombre entre 1 et 100 : 62
Gagné en 3 coups

Exercice 2.6

Écrivez un programme qui demande deux entiers **a** et **b** et qui calcule et affiche la somme des carrés des entiers impairs compris entre **a** et **b** (**a** et **b** inclus).

Par exemple, l'exécution du programme en donnant les valeurs 12 et 256 à **a** et **b** doit produire les affichages suivants :

Donner la valeur de **a** : 12

Donner la valeur de **b** : 256

La somme des carrés des impairs compris entre 12 et 256 vaut : 2795874

Exercice 2.7

Écrivez un programme qui demande deux entiers *a* et *n* et qui affiche les *n* premiers termes de la suite définie par :

$$u_0 = a$$

$$u_i = \frac{1}{2} \times u_{i-1}, \text{ si } u_{i-1} \text{ est pair}$$

$$u_i = 3 \times u_{i-1} + 1, \text{ si } u_{i-1} \text{ est impair}$$

Utilisation de la boucle for

Lorsque l'on doit traiter des données qui sont organisées en séquence comme les chaînes de caractères, le parcours de chaque élément de la séquence s'effectue par l'instruction d'itération à bornes définies **for**. L'exemple ci-dessous affiche les lettres une par une de la chaîne de caractères *bonjour*.

```
for lettre in "bonjour":  
    print(lettre)
```

*Exercice 2.8

Écrivez un programme qui demande la saisie d'une phrase (une chaîne de caractères) et qui affiche combien de voyelles contient la phrase saisie. On comptera le nombre de voyelles minuscules et majuscules non accentuées.

*Indices : utilisez l'opérateur **in** qui permet de tester si un caractère appartient à une chaîne pour tester si une lettre est une voyelle.*

Exemple d'exécution :

Donner une phrase : Combien cette PHRASE contient-t-elle de VOYELLES ?

La phrase : Combien cette PHRASE contient-t-elle de VOYELLES ?

contient : 17 voyelles.

*Exercice 2.9

Écrivez un programme qui demande la saisie d'une chaîne de caractères et qui :

- vérifie que cette la chaîne entrée code un brin d'ADN, c'est à dire qu'elle n'est composée que des caractères A, C, G et T
- et si c'est le cas, construit et affiche la chaîne qui code le brin complémentaire, c'est à dire la chaîne où chaque A est remplacé par un T (et vice-versa) et chaque C par un G (et vice-versa).

Exemples d'exécution :

Donner une séquence de lettres qui représente un brin d'ADN : AGCTTAAXCAAGCT

Il y a au moins une erreur dans le brin donné

Donner une séquence de lettres qui représente un brin d'ADN : CAGTTAACCAGAT
Le brin complémentaire de :
CAGTTAACCAGAT
est :
GTCAATTGGTCTA

Exercice 2.10

Écrivez un programme qui demande la saisie d'une phrase (une chaîne de caractères) et qui affiche cette phrase en remplaçant toutes les lettres qui ne sont pas des voyelles par le caractère *. Les voyelles doivent donc être affichées telles quelles ainsi que tous les autres caractères (ponctuation, etc ...)

Indices : pensez à utiliser les opérateurs de comparaisons sur les caractères. Une lettre est un caractère compris entre "A" et "Z" ou entre "a" et "z".

Pour que la fonction `print` ne passe pas à la ligne il faut utiliser le paramètre `end=""`

Exemple d'exécution :

Donner une phrase : Voici des MAJUSCULES, des chiffres (par exemple 1 23 4 8.7) et d'autres caracteres!
*oi*i *e* *A*U**U*E*, *e* **i***e* (*a* e*e***e 1 23 4 8.7) e* *'au***e* *a*a***e*e*!

