

INFORMATIQUE - EXERCICES D'ENTRAÎNEMENT - CORRIGÉS

Les énoncés de ce document sont extraits d'une base d'exercices donnés en Sup (MPSI et PCSI) approximativement à la moitié du premier semestre, et portent donc sur les premières notions d'algorithmique "de base" du programme d'informatique de prépa.

1 - BOUCLES FOR

EXERCICE 1. — Ecrire un programme qui affiche 50 fois "Je dois ranger mon bureau".

```
## EXERCICE 1 – Bonne résolution
```

```
for k in range(50):
    print("Je dois ranger mon bureau")
```

```
# EXERCICE 1 – Alternative
```

```
for objectif_prepa in range(50): # Liberté pour choisir le nom du compteur...
    print("Je dois ranger mon bureau")
```

EXERCICE 2. — Calculer la somme $S = 4 + 5 + \dots + 2026$.

```
## EXERCICE 2 – Calcul d'une somme
```

```
S = 0 # On commence par initialiser la somme à 0
for k in range(4,2027): # range(a, b) contient les entiers de a à (b-1)
    S = S + k
print("4+5+...+2026 = ",S) # Affichage du résultat (print(S) aurait suffi)
```

EXERCICE 3. — Calculer la somme $S = 4^2 + 5^2 + \dots + 2026^2$.

```
## EXERCICE 3 – Calcul d'une autre somme
```

```
S = 0 # On commence par initialiser la somme à 0
for Jean_Bart in range(4,2027): # range(a, b) contient les entiers de a à (b-1)
    S = S + Jean_Bart * Jean_Bart
print("4²+5²+...+2026² = ",S)
```

EXERCICE 4. — Calculer la somme $S = 5 + 8 + 11 + \dots + 401$.

```
## EXERCICE 4 – Somme avec pas...
```

```
S = 0 # On commence par initialiser la somme à 0
for k in range(5,402,3): #
    S = S + k
print('5+8+11+...+401 =',S)
```

EXERCICE 5. — Demander à l'utilisateur de saisir un entier n , puis afficher la somme :

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

```
## EXERCICE 5 – Somme et interaction avec l'utilisateur

S = 0 # On commence par initialiser la somme à 0
print('Bonjour! Choisissez un entier naturel non nul')
N = int(input()) # La valeur saisie est stockée dans N
for k in range(1,N+1): # range(a, b) contient les entiers de a à (b-1)
    S = S + 1/k
print('1+1/2+...+1/',N,'= ',S)
```

EXERCICE 6. — Simulation de lancers de dé.

Ecrire un programme qui demande à l'utilisateur de saisir un entier n , et qui affiche n entiers choisis aléatoirement entre 1 et 6.

Remarque. On pourra utiliser la fonction `randint` de Python, en l'important à l'aide de l'instruction :

```
from random import randint
```

Ceci fait, l'instruction `randint(a,b)` (avec a et b entiers) renvoie un entier choisi aléatoirement entre a et b au sens large.

```
## EXERCICE 6 – Simulation de lancers de dé

from random import randint
print('Bonjour! Choisissez un entier naturel non nul')
N = int(input()) # La valeur saisie est stockée dans N
for mpsi_ou_pcsi_les_deux_sont_trop_cool in range(N): # for k in range(N) convient aussi!
    print(randint(1,6),end=' ')
```

2 - CONSTRUCTIONS DE LISTES AVEC L'INSTRUCTION FOR

Deux exemples pour commencer. L'instruction

```
L = [ k**2 for k in range(2,6) ]
```

crée la liste

```
L = [4, 9, 16, 25]
```

Et l'instruction

```
L = [ 2*elem + 1 for elem in [3,7,10,20] ]
```

crée la liste

```
L = [7, 15, 21, 41]
```

A l'aide de ces exemples, répondre aux questions suivantes.

EXERCICE 7. — Ecrire une instruction permettant de créer la liste des multiples de 3 compris entre 1 et 100.

```
## EXERCICE 7 – Multiples de 3

L = [] # Initialisation de la liste
for k in range(3,101,3):
    L = L + [k]
print(L)

# EXERCICE 7 – Alternative

L2 = [M_Lortil for M_Lortil in range(3,101,3)]
print(L2)
```

EXERCICE 8. — **Simulation de lancers de dé.** Ecrire une instruction permettant de créer une liste de 20 entiers choisis aléatoirement entre 1 et 6.

```
## EXERCICE 8 – Simulation de lancers de dé

L = [] # Initialisation de la liste
for compteur in range(20):
    L = L + [randint(1,6)]
print(L)

# EXERCICE 8 – Alternative

L2 = [randint(1,6) for M_Zahnd in range(20)]
print(L2)
```

EXERCICE 9. — Ecrire une instruction permettant de créer la liste :

[0, 1, 0, 1, 0, 1, 0, 1]

```
## EXERCICE 9 – Modulo 2

L = [] # Initialisation de la liste
for k in range(8):
    L = L + [k % 2]
print(L)

# EXERCICE 9 – Alternative

L2 = [M_Momin % 2 for M_Momin in range(8)]
print(L2)
```

EXERCICE 10. — Ecrire une instruction permettant de créer la liste :

[0, 0, 1, 1, 2, 2, 3, 3]

```
## EXERCICE 10 – Partie entière
```

```
L = [] # Initialisation de la liste
```

```
for k in range(8):
```

```
    L = L + [k//2] # k//2: quotient dans la div euclidienne de k par 2
```

```
print(L)
```

```
# EXERCICE 10 – Alternative
```

```
L2 = [M_Pacary // 2 for M_Pacary in range(8)]
```

```
print(L2)
```

3 - INSTRUCTIONS CONDITIONNELLES IF, ELIF, ELSE

EXERCICE 11. — Ecrire un programme qui demande à l'utilisateur de saisir 2 valeurs, et qui affiche la plus petite des 2 valeurs.

```
## EXERCICE 11 – Minimum
```

```
print(' Choisissez deux entiers a et b')
```

```
a = int(input())
```

```
b = int(input())
```

```
if a <= b:
```

```
    print('Le plus petit est:', a)
```

```
else:
```

```
    print('Le plus petit est:', b)
```

EXERCICE 12. — Ecrire un programme qui demande à l'utilisateur deux entiers pairs a et b , et qui affiche la moyenne de ces deux entiers.

Si au moins un de ces entiers est impairs, le programme doit afficher un message d'erreur.

```
## EXERCICE 12 – Minimum
```

```
print(' Choisissez deux entiers PAIRS a et b')
```

```
a = int(input())
```

```
b = int(input())
```

```
if a%2!=0 or b%2!=0:
```

```
    print('Erreur de saisie . Au revoir.')
```

```
else:
```

```
    print('La moyenne est:', (a+b)/2)
```

EXERCICE 13. — Ecrire un programme qui calcule les 20 premiers termes de la table de multiplication par 11, mais n'affiche que ceux qui sont des multiples de 3.

```
## EXERCICE 13 – Table de multiplication à trous
```

```
for k in range(1,21):
    if (11 * k) % 3 ==0:
        print('11 x ',k, ' = ',11*k)
```

EXERCICE 14. — Ecrire un programme qui demande à l'utilisateur de saisir 2 chaînes de caractères, et qui affiche la plus grande des 2 chaînes (celle qui a le plus de caractères).

Si les deux chaînes sont de même longueur, le programme doit afficher 'match nul'.

```
## EXERCICE 14 – Chaînes de caractères
```

```
print(' Saisissez deux chaînes de caractères ')
CHAINE1 =input()
CHAINE2 =input()
if len(CHaine1) >len(CHaine2):
    print('La chaîne la plus longue est:', CHAINE1)
elif len(CHaine1) ==len(CHaine2):
    print('Match nul')
else:
    print('La chaîne la plus longue est:', CHAINE2)
```

4 - FONCTIONS

EXERCICE 15. — Ecrire le code Python d'une fonction `fonc1(x)` qui prend comme paramètre un flottant x , et qui retourne $x^2 + x + 1$.

```
## EXERCICE 15 – Fonction
```

```
def fonc1(x):
    return x**2 +x +1
```

EXERCICE 16. — Ecrire le code Python d'une fonction `NB4(L)` qui reçoit comme paramètre une liste d'entiers L , et qui renvoie le nombre d'apparitions de l'entier 4 dans la liste L .

Par exemple l'instruction `NB4([1,2,4,5,7,4,12])` doit produire le résultat : 2

```
## EXERCICE 16 – Nombre de 4
```

```
def NB4(L):
    nombre_de_4 =0
    for element in L:
        if element ==4:
            nombre_de_4 +=1
    return nombre_de_4
```

EXERCICE 17. — Ecriture décimale.

Construire une fonction `Deci(n)` qui reçoit comme paramètre un entier n , et qui retourne la liste de ses chiffres dans son écriture décimale.

Par exemple `Deci(453)` doit renvoyer `[4,5,3]`.

```
## EXERCICE 17 – Ecriture décimale
```

```
def Deci(n):  
    List_Chiff = []  
    while n > 0:  
        chif = n % 10  
        List_Chiff = [chif] + List_Chiff  
        n = n // 10  
    return List_Chiff
```

EXERCICE 18. — Ecriture binaire.

Construire une fonction `Bin(n)` qui reçoit comme paramètre un entier n , et qui retourne son écriture binaire.

Par exemple `Bin(13)` doit renvoyer `1101`.

```
## EXERCICE 18 – Ecriture binaire
```

```
def Bin(n):  
    EcriBin = ''  
    while n > 0:  
        digit = n % 2  
        EcriBin = str(digit) + EcriBin  
        n = n // 2  
    return int(EcriBin)
```