

TD L3 MF – Programmation en Python

TD 1	2
EXERCICE 1	2
EXERCICE 2	2
EXERCICE 3	2
EXERCICE 4	2
EXERCICE5	2
TD 2	3
EXERCICE 6	3
EXERCICE 7	3
EXERCICE 8	3
EXERCICE 9	3
EXERCICE 10	4
TD 3	5
EXERCICE 11	5
EXERCICE 12	5
EXERCICE 13	5
TD 4	6
EXERCICE 14	6
TD 5	7
EXERCICE 15	7
TD 7	8
EXERCICE 16	8
TD 8 - IMPORTS/EXPORTSPYTHON-EXCEL	9
EXERCICE 17	9
TD 9	11
EXERCICE 18	11
EXERCICE 19	11
TD 10	12
EXERCICE 20	12
EXERCICE 21	12
EXERCICE 20	12
EXERCICE 21	12
TD 12 - VBA SOUS EXCEL	13
EXERCICE 22	13

Pour installer Python sous windows ou bien sous Mac OS X, se rendre sur la page :
<https://www.python.org/downloads/>

Remarque : Python est déjà installé sur les ordinateurs de l'université.

TD 1

Exercice 1

a) Ecrire un programme qui affiche le texte suivant :

```
j'affiche ce que je veux
```

b) Ecrire un programme qui affiche le texte suivant :

```
utiliser la combinaison suivante
```

```
pour passer a la ligne
```

Exercice 2

Soient les trois variables suivantes :

```
entier = 100  
pi = 3.1416  
mot = "test"
```

Ecrire un programme qui affiche ces 3 variables avec les commentaires suivants :

```
Voici un nombre entier : 100  
3.141600 est un nombre décimal  
test est une chaîne de caractères
```

Exercice 3

Soit les deux variables suivantes :

```
a = 1  
b = 2
```

Ecrire un programme qui affiche les variables a et b, les échange, puis les affiche à nouveau.

Exercice 4

Ecrire un programme qui affecte une valeur à un taux de TVA, une valeur à un montant TTC puis calcule le montant HT correspondant.

Exercice 5

a) Testez le programme suivant :

```
prenom = "Nicolas"  
age = 38  
taille = 1.8  
  
print("Ton prénom est", prenom)  
print("Ton âge est", age)  
print("Ta taille est", taille, "m")
```

b) Modifiez le pour que les trois variables (utilisez la fonction input())

```
prenom  
age  
taille
```

soient saisies par l'utilisateur.

TD 2

Exercice6

Ecrivez un programme qui permet à l'utilisateur de saisir un nombre, puis affiche si le nombre est pair ou impair. Vous utiliserez pour cela l'opérateur % qui renvoie le reste de la division entière (par exemple, $7 \% 2$ renvoie 1, car 7 divisé par 2 vaut 3, reste 1).

Exemples d'exécution :

```
Entrez un nombre entier :35
votre nombre est impair

Entrez un nombre entier :36
votre nombre est pair
```

Exercice7

Généralisez l'exercice précédent : saisir un nombre entier P et un nombre entier Q pour dire si Q divise P ou non.

Exercice 8

Ecrivez un programme qui permet à l'utilisateur de saisir deux nombres, puis affiche simultanément le maximum et le minimum des deux nombres :

```
Entrez un 1er nombre :
5

Entrez un 2eme nombre :
9

Maximum : 9
Minimum : 5
```

Exercice 9

Ecrire un programme qui permet à l'utilisateur de saisir un nombre entier, puis affiche ses diviseurs. Exemple d'exécution :

```
Entrez un nombre entier positif : 36
1
2
3
4
6
9
12
18
36
```

Exercice 10

a) En vous inspirant de l'exercice précédent, écrivez un programme qui permet de saisir deux nombres entiers M et N, puis affiche tous les diviseurs communs aux deux nombres.

Exemple d'exécution :

```
Entrez un nombre entier positif N : 36
Entrez un nombre entier positif M : 24
1
2
3
4
6
12
```

b) Modifiez votre programme pour qu'il n'affiche que la dernière valeur :

```
Entrez un nombre entier positif N : 36
Entrez un nombre entier positif M : 24
12
```

TD 3

Exercice 11

Soit la suite mathématique suivante :

$$U_0 = 1$$
$$U_k = 2 \times U_{k-1} + 1$$

Ecrire un programme qui calcule puis affiche les 10 premiers termes de la suite à l'aide d'une boucle.

Exercice 12

En vous inspirant de l'exercice précédent, calculez puis affichez le rang k du premier terme de la suite U dont la valeur U_k dépasse 10000.

Exercice 13

Soit l'algorithme suivant :

- saisir deux nombres entiers N et M
- à chaque étape, remplacer le plus grand des deux nombres par la différence entre le plus grand et le plus petit des deux nombres, jusqu'à ce qu'ils soient égaux.

a) Traduisez cet algorithme en python avec saisie des nombres et affichage des valeurs intermédiaires de M et N séparés par une barre verticale. Exemple d'exécution :

```
Entrez un nombre entier positif N : 36
Entrez un nombre entier positif M : 24
N | M
36 | 24
12 | 24
12 | 12
```

b) Le nombre final est le PGCD (Plus Grand Commun Diviseur) des deux entiers. Vous venez de programmer l'algorithme d'Euclide.

Modifiez votre programme pour écrire le résultat final sous cette forme :

```
Entrez un nombre entier positif : 36
Entrez un nombre entier positif : 24
le PGCD de 36 et 24 vaut 12
```

TD 4

Exercice 14

Soit un tableau `t` contenant `n` éléments. Le programme suivant définit le tableau :

```
t = [6, 12, 9, 1, 5, 10, 8, 3, 4, 13]
n = len(t)
```

a) Afficher toutes les valeurs du tableau de la manière suivante :

```
t[0] : 6
t[1] : 12
t[2] : 9
t[3] : 1
t[4] : 5
t[5] : 10
t[6] : 8
t[7] : 3
t[8] : 4
t[9] : 13
```

b) Ecrire un programme qui affiche le maximum du tableau
(ne pas utiliser la fonction `max()` de Python) :

```
le maximum du tableau est : 13
```

c) Adaptez la solution précédente pour afficher le minimum
(ne pas utiliser la fonction `min()` de Python) :

```
le minimum du tableau est : 1
```

d) Adaptez la solution précédente pour afficher l'indice (numéro de la case) du minimum :

```
l'indice du minimum du tableau est : 3
```

e) Affichez la somme de tous les éléments du tableau
(ne pas utiliser la fonction `sum()` de Python) :

```
somme de tous les éléments du tableau : 71
```

TD 5

Exercice 15

Soit le tableau t du TD précédent contenant les n éléments suivant :

```
t = [6, 12, 9, 1, 5, 10, 8, 3, 4, 13]
n = len(t)
```

a) Affichez le nombre de valeurs > 10 :

```
le nombre d'éléments du tableau supérieurs à 10 est : 2
```

b) Affichez le nombre de valeurs paires :

```
le nombre d'éléments pairs du tableau est : 5
```

c) Affichez la moyenne de tous les éléments du tableau :

```
la moyenne des éléments du tableau est : 7,1
```

d) Écrivez un programme python qui trie le tableau t dans l'ordre croissant puis affiche le tableau trié (ne pas utiliser la fonction sorted() de Python) :

```
[1, 3, 4, 5, 6, 8, 9, 10, 12, 13]
```

TD 7

Exercice 16

Soit un tableau à deux dimensions avec n lignes et m colonnes.

Ce tableau contient les nombres suivants :

	Colonne 0	Colonne 1	Colonne 2
Ligne 0	12	1	0
Ligne 1	1	4	1
Ligne 2	12	1	0
Ligne 3	1	4	1
Ligne 4	1	0	0
Ligne 5	1	4	1
Ligne 6	1	0	0
Ligne 7	1	4	1
Ligne 8	1	0	0

Le programme suivant définit ce tableau de taille n x m :

```
mat = [[12,1,0],[1,4,1],[12,1,0],[1,4,1],[1,0,0],[1,4,1],[1,0,0],[1,4,1],[1,0,0]]
n = len(t)
m = len(t[0])
```

a) Affichez le tableau de la manière suivante :

```
mat[ 0 , 0 ] : 12
mat[ 0 , 1 ] : 1
mat[ 0 , 2 ] : 0
mat[ 1 , 0 ] : 1
mat[ 1 , 1 ] : 4
mat[ 1 , 2 ] : 1
mat[ 2 , 0 ] : 12
mat[ 2 , 1 ] : 1
mat[ 2 , 2 ] : 0
mat[ 3 , 0 ] : 1
mat[ 3 , 1 ] : 4
mat[ 3 , 2 ] : 1
mat[ 4 , 0 ] : 1
mat[ 4 , 1 ] : 0
mat[ 4 , 2 ] : 0
mat[ 5 , 0 ] : 1
mat[ 5 , 1 ] : 4
mat[ 5 , 2 ] : 1
mat[ 6 , 0 ] : 1
mat[ 6 , 1 ] : 0
mat[ 6 , 2 ] : 0
mat[ 7 , 0 ] : 1
mat[ 7 , 1 ] : 4
mat[ 7 , 2 ] : 1
mat[ 8 , 0 ] : 1
mat[ 8 , 1 ] : 0
mat[ 8 , 2 ] : 0
```

b) Affichez la somme de toutes les cases :

```
la somme de toutes les cases vaut : 53
```

c) Calculez et affichez la somme des cases des lignes paires :

```
somme lignes paires : 29
```


TD 8 - Imports/ExportsPython-Excel

Pour les imports/exports de données de Python vers Excel (et vice versa), nous utiliserons la librairie openpyxl.

Pour installer cette librairie, lancer une invite de commandes msdos dans le sous-dossier "Scripts" du dossier où vous avez installé Python, puis taper la commande "pip install openpyxl" (cette manipulation n'est pas nécessaire sur les ordinateurs de l'université car la librairie est déjà installée).

Exercice 17

Télécharger le fichier excel.zip sur la page cred.u-paris2.fr/L3MF

En vous inspirant de l'exemple de la page suivante, écrire un programme qui :

1. importe la série de nombres contenue dans la plage B2:C25 du fichier exemple-import-export-fois-10.xlsx
2. multiplie par 10 dans python les données récupérées
3. exporte le résultat (donc la série de données multipliées par 10) vers la plage E2:F25 de la feuille Excel active
4. enregistre le résultat dans un fichier Excel sous le nom exemple-import-export-fois-10-out.xlsx

```

from openpyxl import Workbook
from openpyxl import load_workbook

#ouvrir le fichier excel sample.xlsx (le fichier doit se trouver dans le même
répertoire que le fichier python)
wb = load_workbook(filename = "sample.xlsx")

#charger la feuille excel active dans une variable python
ws = wb.active

#on récupère la valeur de la case B2 puis on l'affiche
a = ws.cell(row=2, column=2).value
print(a)

#on écrit la valeur de la variable b dans la case B1
b = 425.5
ws.cell(row=1, column=2).value = b

#on enregistre le fichier sample.xlsx
wb.save("sample.xlsx")

#ouvrir un nouveau fichier excel
wb2 = Workbook()

#charger la feuille excel active dans une variable python
ws2 = wb2.active

#on écrit la valeur de la variable b dans la case A2
ws2.cell(row=1, column=2).value = b

#on enregistre le nouveau fichier excel en lui donnant le nom sample2.xlsx
wb2.save("sample2.xlsx")

```

TD 9

Exercice 18

Définissez dans Excel deux matrices quelconques m_1 et m_2 , toutes les deux de même taille, puis importez dans Python ces deux matrices dans deux tableaux à deux dimensions. Calculez ensuite dans Python la matrice m_3 , résultat de la somme des matrices m_1 et m_2 , puis exportez la matrice m_3 dans le fichier Excel de départ.

Exercice 19

Définissez dans Excel une matrice quelconque m , puis importez dans Python cette matrice dans un tableau à deux dimensions. Calculez ensuite dans Python la matrice m_{trans} , transposée de m , puis exportez m_{trans} dans un nouveau fichier Excel.

TD 10

Exercice 20

Programmez l'algorithme suivant de la racine carrée :

```
Saisir une valeur entière N
Rac <- 0
tant que (rac+1)*(rac+1) inférieur ou égale N
    rac <- rac + 1
afficher la valeur de la racine entière de N
```

Exercice 21

Soit X un entier quelconque. Soit la suite définie par récurrence :

```
R0 = X
Rn = (Rn-1 + (X / Rn-1)) / 2
```

a) Ecrire un programme qui calcule R₁₅. Votre programme devra permettre à l'utilisateur de saisir en entrée la valeur X entière de son choix, puis devra afficher toutes les étapes intermédiaires de calcul :

```
Entrez un nombre entier positif : 45287
```

```
étape 0 : 45287
étape 1 : 22644.0
étape 2 : 11322.999977919095
étape 3 : 5663.499768173904
étape 4 : 2835.748030273387
étape 5 : 1425.859033466343
étape 6 : 728.8101188604045
étape 7 : 395.4741944682918
étape 8 : 254.99367760456371
étape 9 : 216.2970797051753
étape 10 : 212.8355750675999
étape 11 : 212.80742654413962
étape 12 : 212.80742468250492
étape 13 : 212.80742468250492
étape 14 : 212.80742468250492
```

b) Que se passe-t-il à partir du rang 12 ?

Exercice 22

Modifier le programme de l'exercice précédent pour qu'il s'arrête automatiquement dès que la suite devient stationnaire.

Exercice 23

```
S0 = 0
Sn = Sn-1 + 1/n2
```

a) Ecrire un programme qui calcule S₁₀₀. Votre programme devra afficher toutes les étapes intermédiaires de calcul.

b) La suite a-t-elle l'air de converger ?

c) Modifier le programme pour qu'il s'arrête automatiquement dès que la suite devient stationnaire. Que se passe-t-il ?

d) Modifier le programme pour qu'il calcule la limite de la suite S_n avec une précision de 5 chiffres après la virgule.

TD 12 - VBA sous Excel

Exercice 24

Programmer l'exercice 15 d), l'exercice 16 b), l'exercice 18 et l'exercice 20 en VBA.