TD L3 EMI – Programmation en Python

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 programmequi 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 programmequi affecte une valeur à un taux de TVA, une valeur à un montant TTC puis calculent le montant HT correspondant.

Saisies de données

Exercice5

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.

Exercice 6

Le symbole % dans l'expression x % ycalcule le reste de la division entière de x par y.

Le symbole // dans l'expression x // y calcule le quotient de la division entière de x par y.

Traduisez les deux algorithmes suivants (qui calculent la date de Pâques) en python, puis testez-les sur l'année 2019 :

Algorithme de Thomas O'Beirne

```
Soit M l'année du calcul (prenons 2005 pour exemple) :
•On pose n = M - 1900 (on retranche 1900 à l'année, donc n = 105 pour notre
exemple)
•On prend a, le reste de n dans la division par 19 (105 / 19 = 5 mais 5 × 19 = 95
au lieu de 105, il reste donc 10 ; a = 10)
•On calcule a × 7 + 1 (ce qui donne pour l'exemple 7 × 10 + 1 = 71)
•On en prend b, le résultat (entier) de la division par 19 (71 / 19 = 3 donc b = 3)
•On calcule (11 × a) - b + 4 (soit 11 × 10 - 3 + 4 = 111)
•On en prend c le reste dans la division par 29 (111 / 29 = 3, or 3 × 29 = 87 au
lieu de 111, il reste donc 111 - 87 = 24, donc c = 24)
•On calcule d la partie entière de n / 4 (105 / 4 = 26)
•On calcule n - c + d + 31 (soit 105 - 24 + 26 + 31 = 138)
•On en prend e le reste dans la division par 7 (138 / 7 = 19, or 19 × 7 = 133 au
lieu de 138, il reste donc 138 - 133 = 5, donc e = 5)
•On calcule P = 25 - c - e (dans l'exemple : P = 25 - 24 - 5 = -4)
•La date de Pâques tombe P jours après le 31 mars (ou avant si P est négatif).
Ce qui signifie que :
- pour P = 1, le 1er avril, autrement dit P positif correspond directement au jour
du mois d'avril
- pour P = 0 le jour de Pâques est le 31 mars, et
- pour P = −1 le 30 mars, autrement dit P négatif doit être ajouté à 31 pour
obtenir le jour du
mois de mars
(Pour l'année 2005, on trouve P = -4, ce qui veut dire que Pâques est le dimanche
31 - 4 = 27 mars).
```

Algorithme de Oudin

```
On prendra pour exemple le calcul dans l'année 2009. Les divisions doivent toujours
être entières (on supprime les décimales).
•G qui représente le nombre d'or diminué de 1: Diviser l'année par 19, en prendre
le reste
( 2009/19=105 or 105x19=1995 et il nous faut 2009, donc l'écart vaut G=14 )
•C et C_4 permettent le suivi des années bissextiles: diviser l'année par 100 puis
encore par 4
( 2009/100=C=20 et 20/4=C_4=5 )
•E : Diviser (8 x C + 13) par 25 sans les décimales
( 8x20+13=173 /25=E=6 )
•H qui dépend de l'épacte : diviser (19xG + C - C_4 - E + 15) par 30, en prendre le
reste
( On prend le reste d'une division selon le même principe que pour G: (290)/30=9 or
9x30=270 et il
nous faut (290), donc l'écart vaut H=20 )
•K : diviser H par 28
( 20 / 28 = K = 0 )
•P : diviser 29 par (H+1)
( 29 / 21 = P = 1 )
•Q : diviser (21-G) par 11
( 21-14=7 /11=Q=0 )
•I représente le nombre de jours entre la pleine lune pascale et le 21 mars : (
KxPxQ - 1 ) x K + H
( 0x1x0-1=-1 x0=-0 + 20= I=20 )
•B : diviser l'année par 4 et enlever les décimales, y ajouter l'année
( 2009/4=502 +2009 = 2511 )
•J1 : Additionner B + I + 2 + C_4 et retrancher C
( J1 = 2518 )
•J2 calcule le jour de la lune pascale (0=dimanche 1=lundi...6=samedi) : diviser J1
par 7 et en prendre le reste. (On calcule toujours le reste d'une division selon le
même principe qu'avec G et H, le résultat est J2=5)
•R le résultat final, enfin : 28 + I - J2
(R= 43)
R représente la date du mois de mars, s'il dépasse 31 on déborde sur avril (30
correspond au 30 mars, 31 au 31 mars, 32 au 1er avril, 33 au 2 avril, ...).
Retrancher 31 le cas échéant pour obtenir la date d'avril. (Pâques 2009 tombe donc
le 12 avril.)
```

TD 2

Structures conditionnelles

Exercice 7

Ecrivez un programme qui permet à l'utilisateur de saisir un nombre, puis affiche si le nombre est pair ou impair. Exemples d'exécution :

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

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

Exercice 8

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

Exercice 9

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 10

Ecrire un programme qui permet à l'utilisateur de saisir un nombre et d'indiquer s'il est ou non dans l'intervalle [0,20]. Exemple d'exécution :

```
Tapez un nombre entier : 15

15 appartient à l'intervalle [0,20]
```

Structures répétitives

Exercice11

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 12

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
```

Exercice 13

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 14

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

Exercice 15

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 24vaut 12
```

TD 3

Exercice 16

Soit un tableau tcontenantn éléments. Le programme suivant définit letableaut :

```
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'indicedu 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
```

f) Affichez le nombre de valeurs > 10 :

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

g) Affichez le nombre de valeurs paires :

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

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

```
lamoyenne des éléments du tableau est : 7,1
```

i) 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 4

Exercice 17

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(mat)
m = len(mat[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
```

d) Définissez deux matrices m1 et m2 toutes les deux de même taille. Vous utiliserez pour cela deux tableaux à deux dimensions. Vous calculerez ensuite la matrice m3, résultat de la somme des matrices m1 et m2. Vous afficherez ensuite m3.

e) Définissez une matrice m à l'aide d'un tableau à deux dimensions. Calculez dans la matrice mtrans la transposée de m. Vous afficherez ensuite mtrans.

TD 5

Exercice 18

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
afficherla valeur de la racine entière de N
```

Exercice19

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

$R_0 = X$

$R_n = (R_{n-1} + (X / R_{n-1})) / 2$

a) Ecrire un programme qui calcule R_{15} . 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 partirdu rang 12 ?

Exercice 20

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

Exercice 21

$S_0 = 0$
$S_n = S_{n-1} + 1/n^2$

a) Ecrire un programme qui calcule S_{100}. 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 ?

d) Modifier le programme pour qu'il calcule la limite de la suite S_n avec une précision de5 chiffres après la virgule.

Imports/Exports Python-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 22

Télécharger le fichier excel.zip sur la page cred.u-paris2.fr/L3EMI

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 dela 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")
```