Université d'Evry Val d'Essonne
L1 SPI - Année universitaire 2018/2019

Modélisation et Simulation - EC222

TP1 - Premiers programmes en Python

Consignes :

- répondre à chacun des exercices dans un fichier spécifique,
- pour toute question ou remarque : gaetan.hello@univ-evry.fr

1 Exercices

1.1 Variables entières

```
a=2
b=a+3
```

Listing 1 – Variables entières

Compléter le code précédent en définissant une variable c valant la somme de a, b et 4

1.2 Variables flottantes et conversion entière

```
a=1/2
b=int(3.8)
```

Listing 2 – Variables flottantes

Compléter le code précédent en définissant une variable c dont la valeur est telle que la partie entière de la somme entre a, b et c vaille 4. A quelle plage de valeurs appartient c.

1.3 Chaînes de caratères

```
message="mon prenom est "
monNom="Albert"
message=message+monNom
print(message)
```

Listing 3 – Chaînes de caratères

Rédiger un script qui stockera votre nom et votre prénom dans deux variables (attention aux caractères accentués) et qui affichera un message vous présentant.

1.4 Opérateurs de comparaison

```
a=2
b=3
test1=(a==b)
test2=(a!=b)
test3=(a>b)
test4=(a>=b)
test5=(a>=b)and(a<10)
test6=(a>=b)or(a<10)
print(test1,test2,test3,test4,test5,test6)
```

Listing 4 – Opérateurs de comparaison

Ecrire un script qui affichera si 3 nombres a,b et c sont tels que a>=b>=c ou c>=b>=a.

1.5 Structures conditionnelles

```
a=2
b=3
if a>b:
  message="a>b"
elif a<b:
  message="a<b"
else:
  message="a=b"
print(message)
```

Listing 5 – Structures conditionnelles

Ecrire un script qui affichera la mention obtenue au bac pour une moyenne donnée.

1.6 Boucles for

```
#calcul de la somme des n premiers nombres entiers
n=10
somme=0
for i in range(n):
    somme=somme+i+1
print(somme)
```

Listing 6 – Boucles for

Calculer la somme des carrés des n premiers nombres entiers.

1.7 Boucle for

```
#calcul de la moyenne des valeurs de la liste notes
notes=[11, 9, 5, 15, 17]
nbNotes=len(notes)
somme=0
for i in range(nbNotes):
    somme=somme+notes[i]
moyenne=somme/nbNotes
print(moyenne)
```

Listing 7 – Boucle for

Un étudiant a obtenu les notes suivantes dans cinq matières différentes : 11, 9, 5, 15 et 17. Chaque matière possède un coefficient valant respectivement 8, 6, 6, 4 et 4. Compléter alors le code précédent pour calculer la moyenne pondérée des notes.

1.8 Boucles while

```
#duree necessaire au doublement de son livret A
somme0=1000
taux=0.75/100
somme=somme0
n=0
while somme<2*somme0:
  somme*=(1+taux)
  n+=1
print(n)
```

Listing 8 – Boucles while

Un gateau pèse 1kg. On souhaite le partager en parts égales de masse inférieure à 200g. Quels sont alors le nombre de parts maximal et le nombre de découpes minimal qu'il faut réaliser ? (chaque découpe partageant en 2 la part coupée)

1.9 Modules math, numpy, matplotlib

```
import math
import numpy
import matplotlib.pyplot as pyplot
pyplot.close()
n=100
x=numpy.zeros(n)
y=numpy.zeros(n)
for i in range(n):
    x[i]=4*math.pi/(n-1)*i
    y[i]=math.cos(x[i])
pyplot.plot(x,y)
```

Listing 9 – Modules

Tracer le cercle de centre (0,0) et de rayon 1.

1.10 Fonctions

```
def AireTriangle(b,h):
  aire=b*h/2
  return aire

base=10
hauteur=13
aire=AireTriangle(base, hauteur)
print(aire)
```

Listing 10 – Fonctions

Définir une fonction qui retourne la valeur maximale entre ses deux arguments réels.

2 Problèmes

2.1 Estimation de π par la formule de Leibniz

Il existe de nombreuses méthodes pour estimer la valeur du nombre π. En particulier, la représentation de π par des séries (sommes infinies) permet d'accéder à une estimation au moyen de programmes simples à coder. La formule de Leibniz exprime ainsi π selon :

$$\frac{\pi}{4} = \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1} \tag{1}$$

A/ Rédiger un script python permettant d'estimer π au moyen de la formule précédente pour un nombre de termes n donné. Quelles estimations obtient-on pour $n = 5, 10, 20, 40, 80$ termes ? (4 pt)

B/ On souhaite estimer π avec une précision relative de 10^{-3}. Python possède la variable *math.pi* qui servira ici de référence. Modifier le programme précédent (indice : une boucle *while* semble ici adaptée) de sorte à évaluer le nombre de termes nécessaires. (4 pt)

C/ Plus la série tronquée possède de termes, plus celle-ci conduit à une estimation précise de π. On se propose ainsi d'observer la convergence de la méthode. Pour cela, les valeurs intermédiaires de la somme et les erreurs relatives seront stockées dans des tableaux. On représentera ensuite graphiquement l'évolution de la somme intermédiaire et de l'erreur relative en fonction du nombre d'itérations réalisé. Commenter. (6 pt)

2.2 Trajectoire et vitesses d'un mobile

Un boulet de canon de masse m=10 kg est projeté du sommet du mont Everest (y_0=8848 m) à la vitesse v de 100 m/s selon un angle α=30°. Le projectile est soumis à l'accélération de la pesanteur terrestre g=9.81 m/s^2 ainsi qu'au frottement de l'air représenté par le modèle de Stokes avec k=1 N.s/m.

A/ Représenter la trajectoire du mobile jusqu'à T=100 s après l'instant initial. (4 pt)

B/ Représenter l'évolution des composantes de vitesse du mobile sur l'intervalle de temps $[0, T]$. (4 pt)

3 BONUS : Enveloppe de tir

Un boulet de canon de masse m=10 kg est projeté du point de coordonnées $(x_0, y_0) = (0, 0)$ à la vitesse v selon un angle α. Le projectile est soumis à l'accélération de la pesanteur terrestre g=9.81 m/s^2 ainsi qu'au frottement de l'air représenté par le modèle de Stokes avec k=1 N.s/m. Une cible rectangulaire $[400, 420] \times [0, 30]$ est visée.

3.1 Pour $(v, \alpha) \in [50, 200] \times [0, \pi/2]$, représenter les trajectoires atteignant la cible.

3.2 Pour $(v, \alpha) \in [50, 200] \times [0, \pi/2]$, représenter le sous-domaine de l'espace (v, α) où la cible est atteinte.