

Programme 1 : tirage aléatoire d'un ordre de passage.

```
from random import *
noms=["toto","titi","toutou"]
result=[]

for i in range(len(noms)) :
    a=choice(noms)
    noms.remove(a)
    result.append(a)
    print ("semaine",i+1,":",a)

print (result)
```

Programme 2 : Saisie de valeur et affichage d'une courbe.

```
from math import *          #importe toutes les instructions du module math
import matplotlib.pyplot as plt  #importe toutes les instructions du module matplotlib et les appelle par plt.
import numpy as np
import scipy.stats as sc

print("* Entraînement au tracé de courbe sur python *")
print("Attention : le séparateur décimal est le point")

listec=[] #liste des données de concentration
listeA=[] #valeurs correspondante de la l'Absorbance

print("nombre de point? :") #demande combien de point de mesure
Nb=int(input())          # demande une saisie et la convertie en entier

plt.plot(listec,listeA_model,color='red')
plt.title("Absorbance A en fonction de la concentration C (en mol/L)")
plt.xlabel("C(en mol/min)")

plt.ylabel("Absorbance A ")

plt.show()
```

Programme 3 : Trajectoire d'une pierre.

```
import numpy as np
import matplotlib.pyplot as plt    # Appel des fonctions et leur donne un nom court

V0x = float(input(" Entrez la valeur de la vitesse horizontale initiale"))
V0y = float(input(" Entrez la valeur de la vitesse verticale initiale "))
X0=float(input(" Entrez la valeur de l'abscisse de départ en mètre"))
Y0=float(input(" Entrez la valeur de la hauteur (ordonnée)de départ en mètre"))

#Positions du mobile
t=np.linspace(0.0,3,30)           # t prend 30 valeurs entre 0 à 3 s
x= V0x*t+X0                       # abscisses des positions (on accepte ici l'expression)
y= Y0 +V0y*t-9.8*t*t             # ordonnées des positions (idem)

# Tracé des points de la trajectoire
plt.axis([0,12,0,15])             # impose la plage du repère de 0 à 15m pour les x et de 0 à 15m pour les y
plt.plot(X0,Y0,'go',ms=6)        # dessine un gros rond vert aux coordonnées x=X0 y=Y0 ;
plt.text(X0,Y0, 'A', horizontalalignment = 'center', verticalalignment = 'bottom')
# place sous la position X0 Y0 le texte 'A'

plt.plot(x,y,'ro',markersize=2)  # trace les points avec une taille de 2 sous forme de rond rouge.
plt.plot(x,y)                    # trace la courbe en bleu.
plt.xlabel("x (en m)")           # légende des abscisses
plt.ylabel("y (en m)")           # légende des ordonnées
plt.title("il faut un titre")     # titre
plt.show()                       #affiche la fenêtre du graphique
```

I) Modifier le programme pour :

- 1) Mettre un titre au graphique.
- 2) Ajouter dans le tracer du graphique les points de passage obligés de la pierre sous forme de rond rouge.
 - a. Le point B en 4,11
 - b. Le point C en 5.2,11.4
 - c. Le point D en 9,11
- 3) Ajouter sur le graphe les noms des points B, C et D.

II) Utiliser le programme pour :

- 4) Chercher des valeurs de V_x (composante horizontale de la vitesse de départ) et V_y (composante verticale de la vitesse de départ) et les coordonnées X_0 et Y_0 qui permettent de faire passer la trajectoire par les points B, C et D

Programme 4 : enregistrer un fichier .csv

```
from tkinter import *
from tkinter.filedialog import askopenfilename
import matplotlib.pyplot as plt

root = Tk()
# on peut donner le chemin d'accès vers le fichier ou la fonction askopenfilename
name = askopenfilename(parent=root, filetypes=(("Fichier CSV", "*.csv"),
        ("Fichier Texte", "*.txt"), ("Tous les fichiers", "*.*")),
title="Choisir un fichier")
root.destroy()

sep = ";" # caractere separateur du csv -peut etre une virgule, un point-virgule
ou une tabulation notée \t
entete = 0 # nombre de lignes d'entete

f = open(name, "r")
# on lit toutes les lignes et on met ça dans une liste -un élément par ligne-
data = f.readlines()
f.close() # on referme le fichier

# on supprime les lignes d'en-tête qui ne nous intéressent pas
data = data[entete:]

# on construit les listes de valeurs utiles
liste1 = []
liste2= []
# liste3 = []
# liste4 = []

for ligne in data:
    # change les virgules en point => format numérique différent sur excel et sur
python
    ligne = ligne.replace(",", ".")
    # on sépare les différents éléments en utilisant le caractère séparateur défini
    ligne = ligne.strip().split(sep)
    ligne = list(map(float, ligne)) # on convertit chaque élément en float
    # on rentre les valeurs dans les listes adaptées
    liste1.append(ligne[0])
    liste2.append(ligne[1])
    # liste3.append(ligne[2])
    # liste4.append(ligne[3])

print("Les données sont en mémoire dans des listes python")
```

Programme 5 : Tracé la courbe d'un fichier .csv

```
from tkinter import *
from tkinter.filedialog import askopenfilename
import matplotlib.pyplot as plt

root = Tk()# on peut donner le chemin d'accès vers le fichier ou la fonction askopenfilename
name = askopenfilename(parent=root, filetypes=(("Fichier CSV", "*.csv"),
        ("Fichier Texte", "*.txt"), ("Tous les fichiers", "*.*")),
title="Choisir un fichier")
root.destroy()

sep = ";" # caractere separateur du csv -peut etre une virgule, un point-virgule ou une tabulation notée \t
entete = 0 # nombre de lignes d'entete

f = open(name, "r")
# on lit toutes les lignes et on met ça dans une liste -un élément par ligne-
data = f.readlines()
f.close() # on referme le fichier

# on supprime les lignes d'en-tête qui ne nous intéressent pas
data = data[entete:]

# on construit les listes de valeurs utiles
liste1 = []
liste2= []
# liste3 = []
# liste4 = []

for ligne in data:
    # change les virgules en point => format numérique différent sur excel et sur python
    ligne = ligne.replace(",", ".")
    # on sépare les différents élément en utilisant le caractère séparateur défini
    ligne = ligne.strip().split(sep)
    ligne = list(map(float, ligne)) # on convertit chaque élément en float
    # on rentre les valeurs dans les listes adaptées
    liste1.append(ligne[0])
    liste2.append(ligne[1])
    # liste3.append(ligne[2])
    # liste4.append(ligne[3])

print("Les données sont en mémoire dans des listes python")

plt.plot(Lambda, A, "+:", color="green", label="A = f(lambda)")
plt.legend()

# Textes sur les axes
plt.xlabel("Lambda (nm)")
plt.ylabel("Absorbance")

# Permet de choisir les limites des deux axes
#ax = plt.gca( )
# ax.set_xlim(0.,2.)
# ax.set_ylim(0.,5.)

# Permet de choisir où seront les valeurs le long des axes
#plt.xticks([0.25*i for i in range(9)])
#plt.yticks([i for i in range(6)])

# Afficher une grille (au niveau des valeurs des axes)
plt.grid()

plt.title("Spectre d'absorption de Mon")
plt.show()
```

