

Devoir maison - Informatique

CORRIGÉ

I - Exercices en vrac

1. Passage à un

```
#---- fonction de complexité linéaire en len(L)
def passage_a_un(L) :
    for k in range (len(L)) :
        if L[k]==1 : return k

#---- fonction de complexité logarithmique en len(L)
def passage_a_un_FAST(L) :
    mil = len(L)//2
    while True :
        if L[mil]==0 : #----- L[mil]==0
            if L[mil+1]==1 : return mil+1
            mil = (mil+len(L)) // 2
        else : #----- L[mil]==1
            if L[mil-1]==0 : return mil
            mil = mil//2
```

2. Défoncer une ligne

```
from random import *
a      = [ [1]*10 for i in range(10) ]
sum     = 10
NbCoups = 0
while sum>0 :
    line = randint(0,9)
    a[line][randint(0,9)] = 0
    sum = 0
    for k in a[line] : sum += k
    NbCoups += 1
print(NbCoups)
```

3. Minimum local

```
alt = [ [i*i + i*j - 60*i + 1.2*j*j - 90*j + 2180 for j in range(100) ] for i in range(100) ]

def Deplace(i,j):
    for k in range(-1,2):
        for m in range(-1,2):
            if alt[i+k][j+m]<alt[i][j] :
                return( [i+k , j+m] )
    return(-1)

Pos = [50,50]
while Deplace(Pos[0], Pos[1]) != -1:
    Pos = Deplace(Pos[0], Pos[1])
    # print(Pos , alt[Pos[0]][Pos[1]]) # si on veut voir le trajet
print (Pos) # affiche [14,32]
```

4. Le bébé fou sur l'échiquier

```
TI = 8
TJ = 13
a = [ [ 0 for j in range(TJ) ] for i in range(TI) ]

i = 0 # position du fou (numéro de ligne)
j = 0 # position du fou (numéro de colonne)
vi = 1 # vitesse selon les lignes
vj = 1 # vitesse selon les colonnes
a[0][0] = 1

while a[i][j] != 3:
    i = i + vi
    j = j + vj
    if i==0 or i==TI-1 : vi = -vi
    if j==0 or j==TJ-1 : vj = -vj
    a[i][j] = a[i][j] + 1

print(i,j) for x in a : print(x)    # pour afficher le tableau
C'est la case de coordonnées (1,11) que le bébé-fou visite pour la troisième fois en premier.
```

5. Du Euler gratos

```
import numpy as np
import matplotlib.pyplot as plt

n    = 1000
y    = np.zeros(n+1)
y[0] = 1

for k in range(n) :
    h = 10 / n    # le pas = "10 divisé par le nombre d'intervalles"
    t = k*h       # l'instant au bout de k pas
    y[k+1] = y[k] + h * (sin(t)-sqrt(t*t+1)*y[k])

plt.plot(np.linspace(0,10,len(y)) , y)
plt.show()
```

II - Euler - Système de deux masses reliées par un ressort

```
import numpy as np
import matplotlib.pyplot as plt
from math import *

d0 = 1 # longueur au repos des ressorts
kRessort = 10 # constante de raideur des ressorts

def Norme(M) : return sqrt(M[0]**2 + M[1]**2)

def Unit(M) : return M/Norme(M)

def F1sur2(M1,M2) :
    d = Norme(M2-M1)
    if d==0 : return np.array([0,0])
    U = Unit(M1-M2)
    return U * (d-d0) * kRessort

def Euler(M1,M2,V1,V2,m1,m2,duree,nbpas) :
    M1Res = np.zeros( (nbpas+1,2) )
    M2Res = np.zeros( (nbpas+1,2) )
    V1Res = np.zeros( (nbpas+1,2) )
    V2Res = np.zeros( (nbpas+1,2) )
    M1Res[0] , M2Res[0] , V1Res[0] , V2Res[0] = M1, M2, V1, V2
    pas = duree / nbpas
    for k in range(nbpas) :
        V1Res[k+1] = V1Res[k] + pas * (1/m1*F1sur2(M2Res[k],M1Res[k]))
        V2Res[k+1] = V2Res[k] + pas * (1/m2*F1sur2(M1Res[k],M2Res[k]))
        M1Res[k+1] = M1Res[k] + pas * V1Res[k]
        M2Res[k+1] = M2Res[k] + pas * V2Res[k]
    return M1Res,M2Res

M1 = np.array ( [0,2] )
M2 = np.array ( [0,0] )
m1,m2 = 1,1
M1R,M2R = Euler(M1,M2,[0.1,0],[0.1,0],m1,m2, 6,1000)
#M1R,M2R = Euler(M1,M2,[-0.1,0],[0.1,0],m1,m2, 6,1000)
#M1R,M2R = Euler(M1,M2,[-1,0],[1,0],m1,m2, 10,10000)
plt.plot(M1R[:,0],M1R[:,1])
plt.plot(M2R[:,0],M2R[:,1])
plt.show()
```

III - Images et compression

```
import matplotlib.pyplot as plt

def uncompress(L) :
    p,n = L[0] , L[1] # p=largeur=L[0] , n=hauteur=L[1]
    act = L[2]         # couleur actuelle
    i,j = 0,0          # coordonnée actuelle du pixel
    res = [ [0]*p for i in range(n) ]
    for k in range(3,len(L)) :
        for x in range(L[k]) :
            res[i][j] = act      # on colorie le pixel avec la couleur actuelle
            j += 1               # on avance d'une case
            if j==p :            # si on arrive en bout de ligne
                j = 0            # on passe à la ligne suivante
                i += 1
        act = 1-act              # on inverse la couleur
    return res

def LireFichier() :
    f = open("moijamais.txt","r")
    C = []
    for s in f : C.append(int(s))
    f.close()
    return C

C = LireFichier()
U = uncompress(C)
plt.imshow(U,cmap="gray")
plt.show()
```