Bilan de l'informatique : Ta fiche!

1. Terme général d'une suite récurrente d'ordre 1

1.1 avec entrée

Voici un programme renvoyant u_n avec $(u_n)_{n \in \mathbb{N}}$ définie par : $u_0 = 12$ et :

$$\forall n \in \mathbb{N}, u_{n+1} = 5u_n^2 + n$$

```
def u(n):  
    u = 12  
    for k in range(n):  
        u = 5 * u**2 + k  
    return (u)
```

1.2 sans entrée, sans sortie

Voici un programme renvoyant u_n avec $(u_n)_{n \in \mathbb{N}}$ définie par : $u_0 = 12$ et :

$$\forall n \in \mathbb{N}, u_{n+1} = 5u_n^2 + n$$

```
def u():  
    u = 12  
    n = eval(input("quel rang veux-tu?"))  
    for k in range(n):  
        u = 5 * u**2 + k  
    print (u)
```

2. Terme général d'une suite récurrente d'ordre 2

Voici un programme renvoyant u_n avec $(u_n)_{n \in \mathbb{N}}$ définie par : $u_0 = 12, u_1 = 7$ et :

$$\forall n \in \mathbb{N}, u_{n+2} = 3u_n - u_{n+1}$$

```
def u(n):  
    u, v = 12, 7  
    for k in range(n):  
        a = v  
        v = 3 * u - v  
        u = a  
    return (u)
```

3. Liste $[u_0, \dots, u_n]$

Voici un programme renvoyant $[u_0, \dots, u_n]$ avec $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 12$ et :

$$\forall n \in \mathbb{N}, u_{n+1} = 5u_n^2 + n$$

```
def u(n):  
    u = 12  
    L=[u]  
    for k in range(n):  
        u = 5 * u**2 + k  
        L.append(u)  
    return (L)
```

4. Produit des éléments d'un tableau

4.1 Tout le tableau

```
def produit(T):  
    p=1  
    nL, nC= size(T,0), size(T,1)  
    for k in range(nL):  
        for j in range(nC):  
            p=p*T[k][j]  
    return (p)
```

4.2 La première colonne

```
def produit(T):  
    p=1  
    nL= size(T,0)  
    for k in range(nL):  
        p=p*T[k][0]  
    return (p)
```

5. Recherche du maximum

5.1 Recherche du maximum dans une liste

```
def max(L):  
    M=L[0]  
    for k in L:  
        if k>M:  
            M=k  
    return M
```

5.2 Recherche du maximum dans un tableau

```
def grand(T):  
    m = T[0,0]  
    for x in T:  
        for y in x:  
            if y > m:  
                m=y  
    return m
```

5.3 Recherche du maximum et son indice

```
def max(L):  
    M, i=L[0], 0  
    n=len(L)  
    for k in range(n):  
        if L[k]>M:  
            M,i =L[k], k  
    return M
```

6. Méthode des rectangles

```
def Rectangles(f, n):  
    R=0  
    for k in range(n):  
        R=R+f(k/n)  
    return R/n
```

7. Faire un graphe

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

xmeth1= np.linspace(0, m.pi, 100)
xmeth2= [k for k in range(n)]
y = [m.cos(m.exp(a)) for a in x]

plt.plot(x, y)
plt.show()
```

8. Algo de Dichotomie

```
def dichotomie(f, a, b, eps):
    while (b-a)>eps:
        u=(a+b)/2
        if f(a)*f(u)<=0:
            b=u
        else:
            a=u
    return u
```

9. Recherche d'un élément dans une liste

```
def recherche(L, x):
    for k in L:
        if k==x:
            return True
    return False
```

10. Le tri par insertion

```
def tri(L):  
    for i in range(1, len(L)):  
        k = L[i]  
        j = i-1  
        while j >= 0 and k < L[j] :  
            L[j + 1] = L[j]  
            j -= 1  
        L[j + 1] = k  
    return L
```

11. Somme-Moyenne-Produit

```
def somme(li):  
    s = 0 #p=1 si produit  
    for x in li:  
        s += x # p=p*x si produit  
    return s #s / len(li) si moyenne
```

12. Étapes pour simuler une variable aléatoire

```
import numpy as np
def de():
    return np.random.randint(1,7)

def Simul(n=1000):
    return [ de ( ) for i in range(n)]

import matplotlib.pyplot as plt

x = Simul()

plt.hist(x,6)

def Moyenne(n=1000):
    a=Simul(n)
    s=0
    for k in a:
        s+=k
    return (s/n)

def Variance(n=1000):
    a=Simul(n)
    s=0
    for k in a:
        s+=k**2
    return (s/n)-(Moyenne(n))**2
```