

```

# ===== #
# PSI* / 2016-2017 #
# DS du 19 novembre #
# ===== #

import numpy as np

## Partie I

def BuildM(La,Lb,Lc):
    n = len(Lb)
    M = np.zeros((n,n))
    # Remplissage de la diagonale de M
    for i in range(n):
        M[i,i] = Lb[i]
    # Remplissage de la diagonale située SOUS la diagonale principale
    for i in range(n-1):
        M[i+1,i] = La[i]
    # Remplissage de la diagonale située AU-DESSUS de la diagonale principale.
    for i in range(n-1):
        M[i,i+1] = Lc[i]
    # Renvoi de la matrice M
    return(M)

## Partie II

def ThomasSolver(M,d):
    # Dimension de la matrice M
    N = M.shape[0]
    # Construction de la liste LcP des c'(i) (i de 0 à N-2)
    LcP = [M[0,1]/M[0,0]]
    for i in range(1,N-1):
        LcP.append(M[i,i+1]/(M[i,i] - M[i,i-1] * LcP[-1]))
    # Construction de la liste LdP des d'(i) (i de 0 à N-1)
    LdP = [d[0,0]/M[0,0]]
    for i in range(1,N):
        LdP.append((d[i,0] - M[i,i-1] * LdP[-1])/(M[i,i] - M[i,i-1] * LcP[i-1]))
    # Construction de la solution
    u = np.zeros((N,1))
    u[-1,0] = LdP[-1]
    for i in range(N-2,-1,-1):
        u[i,0] = LdP[i] - LcP[i] * u[i+1]
    return(u)

# Pour tester...
n = 4
La = [1 for i in range(n-1)]
Lb = [(i+1)**2 for i in range(n)]
Lc = [i+1 for i in range(n-1)]

d = np.array([[ -1],[ 3],[21],[18]])

M = BuildM(La,Lb,Lc)
sol = ThomasSolver(M,d)
print(sol)

## Partie III

"""
A NE PAS DECOMMENTER !
NS = NM = (N-2) + 2*(N-1) + (N-1) = 4N - 5
ND = (N-1) + N = 2N - 1
"""

```

Partie IV

```
"""
A NE PAS DECOMMENTER !
Delta_1 = b
Delta_2 = b^2 - ac
Delta_3 = b^3 - 2abc
"""

def detMn_nr(a,b,c,n):
    x = b
    if n == 1:
        return x
    else:
        y = b**2 - a * c
        for i in range(1,n-1):
            x, y = y, b * y - a * c * x
        return y

def detMn_r(n):
    global a, b, c
    if n == 1:
        return b
    elif n == 2:
        return b**2 - a * c
    else:
        return b * detMn_r(n-1) - a * c * detMn_r(n-2)

"""
# Pour tester...
a = 2
b = 1
c = 3

n = int(input("Donnez la valeur de n : "))
print(detMn_nr(2,1,3,n))
print(detMn_r(n))
"""
```