```python
## Corrigé DS info 2025-26

## Exercice 1
# question 1
def liste_u(n):
    L = []
    u = 1
    for k in range(n+1):
        L.append(u)
        u = -u**2+3*u+5
    return L
# on pouvait aussi faire :
def liste_u_bis(n):
    L = [1]
    u = 1
    for k in range(n):
        u = -u**2+3*u+5
        L.append(u)
    return L

# question 2
import matplotlib.pyplot as plt
absi = [k for k in range(21)]
ordo = liste_u(20)
plt.plot(absi, ordo)
plt.show()
# remarque : en pratique, le code ci-dessus renvoie un message
# d'erreur car les termes de la suite u deviennent rapidement
# trop grands pour que Python puisse les calculer.

# question 3
def nb_pair(L):
    compt = 0
    for x in L:
        if x%2==0:
            compt = compt+1
    return compt

# question 4
def nb_pair_u(n):
    L = liste_u(2*n) # L contient [u_0, u_1, ..., u_2n]
    M = L[n:] # M contient [u_n, u_n+1, ..., u_2n]
    return nb_pair(M)


#%% Exercice 2
# question 1
import numpy as np
N = np.array([[6,7,8],[8,7,6]])

# question 2
def somme_ligne(M,i):
    p = np.size(M,1)
    S = 0
    for j in range(p):
        S = S+M[i,j]
    return S

def somme_colonne(M,j):
    n = np.size(M,0)
    S = 0
```

```
    for i in range(n):
        S = S+M[i,j]
    return S

# question 3
def magique(M):
    a = somme_ligne(M,0)
    n = np.size(M,0)
    p = np.size(M,1)
    for i in range(n):
        if somme_ligne(M,i) != a:
            return False
    for j in range(p):
        if somme_colonne(M,j) != a:
            return False
    return True

# question 4
# La fonction mystere prend en argument une matrice M et, si
# M est magique, calcule la somme S de tous ses coefficients.
# Notant ensuite n le nombre de lignes de M et p son nombre de
# colonne, la fonction renvoie True si S/n = S/p et False sinon.

# Appelons a la somme de chacune des lignes et des colonnes d'une
# matrice magique M de taille (n,p) et supposons que a soit non nul.
# Alors la somme S de tous les coefficients de M vaut d'une part
# n*a (en sommant sur les lignes) et p*a (en sommant sur les colonnes).
# Ainsi les nombres S/n et S/p sont égaux à a et donc S/n = S/p.
# Or comme a est non nul, S est non nul. On peut donc simplifier par S
# et obtenir que n=p, autrement dit M est forcément carrée.

# Il existe des matrices magiques non carrées, mais dans ce cas la somme
# de leurs lignes ou colonnes doit être nulle. Par exemple la matrice
# [[2,-1,-1],[-2,1,1]] est magique sans être carrée.

#%% Exercice 3
# question 1
# On trouve [0,0,1,0,1,1,0,1]

# question 2
def compte(T):
    return sum(T)

# question 3
def evol(T):
    n = len(T)
    L = n*[0]
    for k in range(n-1):
        if T[k]==1:
            if T[k+1]==0:
                L[k]=0
                L[k+1]=1
            else:
                L[k]=1
    return L

# question 4
def vider(T):
    nb_voit = compte(T)
    nb_etapes = 0
    while nb_voit > 0:
```

```
        T = evol(T)
        nb_voit = compte(T)
        nb_etapes = nb_etapes + 1
    return nb_etapes

# question 5
def miroir(L):
    L_bis = []
    n = len(L)
    for k in range(n):
        L_bis.append(L[n-1-k])
    return L_bis
# on pouvait aussi faire :
def miroir_bis(L):
    L_bis = []
    n = len(L)
    for k in range(n):
        x = L.pop()
        L_bis.append(x)
    return L_bis

# question 6
def evol2(T,Tbis):
    L = evol(T)
    M = miroir(Tbis)
    M_next = evol(M)
    L_bis = miroir(M_next)
    return L, L_bis
```