

```
from doctest import testmod

def est_cyclique(plan):
    '''
    Prend en paramètre un dictionnaire `plan` correspondant
    à un plan d'envoi de messages entre `N` personnes A, B, C,
    D, E, F ...(avec N <= 26).
    Renvoie True si le plan d'envoi de messages est cyclique
    et False sinon.

    Tests et Exemples:
    >>> plan_a = {'A':'E', 'B':'F', 'C':'D', 'D':'C', 'E':'B', 'F':'A'}
    >>> est_cyclique(plan_a)
    False
    >>> plan_b = {'A':'C', 'B':'F', 'C':'E', 'D':'A', 'E':'B', 'F':'D'}
    >>> est_cyclique(plan_b)
    True
    '''
    # initialisation de la clé avec la
    # valeur initiale : 'A'
    personne = 'A'
    # longueur du plan
    N = len(plan)
    # inspecter au plus les n-1 successeurs
    for i in range(N - 1):
        if plan[personne] == 'A':
            # un des successeur est 'A', il y a donc au
            # moins 2 cycles
            return False
        else:
            # prochain successeur à explorer
            personne = plan[personne]

    # tous les successeurs ont été exploré
    # et aucun n'a bouclé sur 'A'. Il n'y a donc qu'un cycle
    return True

if __name__ == '__main__':
    testmod()
```