

## Enoncé

On considère au plus 26 personnes A, B, C, D, E, F ... qui peuvent s'envoyer des messages avec deux règles à respecter :

- chaque personne ne peut envoyer des messages qu'à une seule personne (éventuellement elle-même),
- chaque personne ne peut recevoir des messages qu'en provenance d'une seule personne (éventuellement elle-même).

Voici un exemple - avec 6 personnes - de « plan d'envoi des messages » qui respecte les règles ci-dessus, puisque chaque personne est présente une seule fois dans chaque colonne :

- A envoie ses messages à E
- E envoie ses messages à B
- B envoie ses messages à F
- F envoie ses messages à A
- C envoie ses messages à D
- D envoie ses messages à C

Et le dictionnaire correspondant à ce plan d'envoi est le suivant :

```
plan_a = {'A': 'E', 'B': 'F', 'C': 'D', 'D': 'C', 'E': 'B', 'F': 'A'}
```

Un cycle est une suite de personnes dans laquelle la dernière est la même que la première.

Sur le plan d'envoi `plan_a` des messages ci-dessus, il y a deux cycles distincts : un premier cycle avec A, E, B, F et un second cycle avec C et D.

En revanche, le plan d'envoi `plan_b` ci-dessous :

```
plan_b = {'A': 'C', 'B': 'F', 'C': 'E', 'D': 'A', 'E': 'B', 'F': 'D'}
```

comporte un unique cycle : A, C, E, B, F, D. Dans ce cas, lorsqu'un plan d'envoi comporte un *unique cycle*, on dit que le plan d'envoi est *cyclique*.

Pour savoir si un plan d'envoi de messages comportant N personnes est cyclique, on peut utiliser l'algorithme ci-dessous :

- on part d'un expéditeur (ici A) et on inspecte son destinataire dans le plan d'envoi,
- chaque destinataire devient à son tour expéditeur, selon le plan d'envoi, tant qu'on ne « retombe » pas sur l'expéditeur initial,
- le plan d'envoi est cyclique si on l'a parcouru en entier.

Compléter la fonction `est_cyclique` en respectant la spécification.

*Remarque :* la fonction python `len` permet d'obtenir la longueur d'un dictionnaire.

### Script Python

```

1 def est_cyclique(plan):
2     """
3         Prend en paramètre un dictionnaire `plan` correspondant à un plan d'envoi de messages (ici
4         entre les personnes A, B, C, D, E, F).
5         Renvoie True si le plan d'envoi de messages est cyclique et False sinon.
6         """
7     expediteur = 'A'
8     destinataire = plan[ ... ]
9     nb_destinaires = 1
10
11    while destinataire != ...:
12        destinataire = plan[ ... ]
13        nb_destinaires += ...
14
15    return nb_destinaires == ...

```

*Exemples :*

### Script Python

```

>>> est_cyclique({'A':'E', 'F':'A', 'C':'D', 'E':'B', 'B':'F', 'D':'C'})
False
>>> est_cyclique({'A':'E', 'F':'C', 'C':'D', 'E':'B', 'B':'F', 'D':'A'})
True
>>> est_cyclique({'A':'B', 'F':'C', 'C':'D', 'E':'A', 'B':'F', 'D':'E'})
True
>>> est_cyclique({'A':'B', 'F':'A', 'C':'D', 'E':'C', 'B':'F', 'D':'E'})
False

```