

Test python C-Ways - C-Bat'

Objectif

Il faudra écrire un programme capable, à partir d'une distribution d'un jeu de cartes classique, de dérouler une partie de Bataille non classique entre deux joueurs, de désigner le gagnant et de compter le nombre de plis qui ont été faits au cours de la partie.

Règles

Paquet de cartes

Le jeu comporte 52 cartes. Elles sont de 4 couleurs différentes : Spades (Pique), Hearts (Coeur), Diamonds (Carreau) et Clubs (Trèfle).

Il y a 13 valeurs de cartes possibles. Ces valeurs sont ordonnées de la façon suivante, de la carte la plus forte à la carte la plus faible :

Nom (anglais)	Nom (français)
Ace	As
King	Roi
Queen	Dame
Jack	Valet
Ten	Dix
Nine	Neuf
Eight	Huit
Seven	Sept
Six	Six
Five	Cinq
Four	Quatre
Three	Trois
Two	Deux

Si deux cartes ont la même valeur, alors leur force est décidée en fonction de leur couleur. Ainsi :

- Pique bat Coeur et Carreau
- Coeur bat Carreau et Trèfle
- Carreau bat Trèfle
- Trèfle bat Pique

Déroulement d'une partie

Le paquet de 52 cartes est mélangé et distribué aux deux joueurs, Alice et Bob. Chaque joueur a donc 26 cartes dans son paquet.

À chaque tour Alice et Bob placent la carte du dessus de leur paquet sur la table. La carte la plus forte (voir tableau ci-dessus) l'emporte. Par exemple, Alice pose un Roi de Coeur et Bob un Sept de Pique. Le gagnant, ici Alice (car Roi supérieur à Sept), prend alors les cartes sur la table et les met au dessous de son paquet. Alice vient de gagner le pli.

Les joueurs recommencent jusqu'à ce qu'un des deux joueurs ne puisse plus jouer de cartes. Le joueur qui possède toutes les cartes est désigné gagnant.

En cas d'égalité de valeur, par exemple si Alice pose un Huit de Carreau et Bob un Huit de Coeur, il y a bataille. C'est alors la couleur la plus forte qui l'emporte. Dans cet exemple, Coeur bat Carreau et Bob gagne donc le pli.

Précisions

- *Lorsqu'un joueur gagne un pli, il remet les cartes sous son paquet. D'abord la sienne puis celle de l'autre joueur.*
- Si un joueur ne peut plus poser de carte il a perdu.

Exemple d'une partie simplifiée

Voici un exemple d'une partie avec un paquet à 4 cartes : les quatre Rois.

Distribution

Les paquets des joueurs sont distribués de la façon suivante :

```
{
  "Alice": ["King of Spades", "King of Hearts"],
  "Bob": ["King of Clubs", "King of Diamonds"]
}
```

Premier pli

Alice pose le Roi de Pique et Bob la Roi de Trèfle. Bob l'emporte. En effet, les cartes ont la même valeur mais Trèfle bat Pique. Bob met les cartes dans son paquet de la façon suivante : *d'abord la sienne puis celle d'Alice.*

État des paquets à la fin du premier pli :

```
{
  "Alice": ["King of Hearts"],
  "Bob": ["King of Diamonds", "King of Clubs", "King of Spades"]
}
```

Deuxième pli

Alice pose le Roi de Coeur et Bob le Roi de Carreau. La carte d'Alice est cette fois plus forte, puisque Coeur bat Carreau. Alice l'emporte donc. Elle range les 2 cartes ainsi gagnées dans son paquet. *D'abord la sienne, puis celle de Bob.*

État des paquets à la fin du deuxième pli :

```
{
  "Alice": ["King of Hearts", "King of Diamonds"],
  "Bob": ["King of Clubs", "King of Spades"]
}
```

Plis suivants

```
# Pli 3: Hearts beat Clubs

{
  "Alice": ["King of Diamonds", "King of Hearts", "King of Clubs"],
  "Bob": ["King of Spades"]
}

# Pli 4: Spades beat Diamond

{
  "Alice": ["King of Hearts", "King of Clubs"],
  "Bob": ["King of Spades", "King of Diamonds"]
}

# Pli 4: Spades beat Hearts

{
  "Alice": ["King of Clubs"],
  "Bob": ["King of Diamonds", "King of Spades", "King of Hearts"]
}

# Pli 5: Diamonds beat Clubs

{
  "Alice": [],
  "Bob": ["King of Spades", "King of Hearts", "King of Diamonds", "King of Clubs"]
}
```

Dernier pli

Alice pose alors le Roi de Trèfle et Bob le Roi de Carreau. Bob remporte le pli et Alice ne peut alors plus jouer de carte puisque son paquet est vide. Bob l'emporte !

Valeur retournée par le programme

Le programme python devra, à partir de cette distribution, renvoyer le tuple ("Bob", 5). Puisque Bob a gagné en 5 plis.

Rendu

Données fournies en entrée

Un fichier `dealings.json`, sous le format suivant :

```
{ "<PLAYER_A_NAME_STRING>": [ "<CARD_STRING>", "<CARD_STRING>", ... ], "  
<PLAYER_B_NAME_STRING>": [ "<CARD_STRING>", "<CARD_STRING>", ... ] }
```

Programme Python

Il faudra rendre :

- un programme dont le point d'entrée sera nommé `bataille.py`
- un `README.md` qui explique comment lancer le programme et qui contient la réponse trouvée avec le fichier joint à cette énoncé

Critères d'évaluation

- Respect rigoureux des consignes ci-dessus
- Une attention particulière sera portée sur la propreté et la structuration du code
 - *Un code propre n'est pas synonyme d'un code commenté*
- Justesse de la réponse trouvée avec les fichiers joints