Le jeu des skourrou

Le but du projet est de programmer un jeu peu connu : le jeu des skourrou, tout d'abord dans une version débutant, puis dans une version complète avec deux joueurs, enfin une version où un joueur joue contre l'ordinateur.

Pour chaque fonction, on attend une docstring et des asserts.

Première Partie : Présentation du jeu des skourrou pour débutants

Pour jouer aux skourrou, il est nécessaire d'avoir des bouts de bois de trois longueurs différentes. Le nombre de bouts de bois et les règles du jeu varient parfois, nous allons voir dans ce projet plusieurs cas différents.

Question n°1 : Ecrire une fonction creationJeuDebutant(n1,n2,n3) et qui renvoie une liste composée de :

- n1 fois la lettre "G"
- n2 fois la lettre "M"
- n3 fois la lettre "P"

Exemple : creationJeuDebutant(2,3,5) renvoie ["G","G","M","M","M","P","P","P","P","P"]

Remarques :

- "G" désigne un grand bout de bois, "M" un moyen et "P" un petit.
- Dans la suite de l'énoncé, nous ne mettrons plus les guillemets, mais ils seront nécessaires dans le programme Python.

Ce jeu est basé uniquement sur le hasard, le matériel se compose de 18 bouts de bois : 2 grands, 6 moyens et 10 petits. Chaque joueur reçoit 9 bouts de bois de façon aléatoire.

Question n°2 : Ecrire une fonction distribution() qui effectue un partage au hasard du jeu et renvoie les deux listes de bâtons. Il est conseillé d'utiliser la fonction de la question n°1 et la commande choice du module random.

Exemple : distribution() peut renvoyer : ([G,M,M,M,M,P,P,P,P],[G,M,M,P,P,P,P,P,P])

Dans le jeu, le décompte des points se fait de la manière suivante :

- un grand bout de bois rapporte 3 points
- un bout de bois de longueur moyenne rapporte 2 points
- un petit bout rapporte 1 point

Le gagnant est celui qui a le plus de points. En cas d'égalité de points, le gagnant est celui qui a le plus de grands bouts de bois.

Exemples :

- [G,M,M,M,M,P,P,P,P] gagne contre [G,M,M,P,P,P,P,P,P]
- [M,M,M,M,M,M,P,P,P] gagne contre [G,G,P,P,P,P,P,P,P]
- [G,G,M,P,P,P,P,P,P] gagne contre [M,M,M,M,M,P,P,P,P]

Question n°3 :

Ecrire une fonction compteDebutant(L) qui prend en paramètre une liste de bâtons L et renvoie un dictionnaire indiquant le nombre de points et le nombre de bâtons de chaque type dans la liste L.

Exemple : compteDebutant([G,M,M,M,M,P,P,P,P]) renvoie {"points":15 ,"G":1 ,"M":4 ,"P":4}

Question n°4 :

Ecrire une fonction gagnantDebutant(L1,L2) qui affiche :

" le joueur n°1 a gagné" si la liste L1 est plus forte que la liste L2 ou " le joueur n°2 a gagné" si la liste L2 est plus forte que la liste L1 ou " match nul" sinon.

Deuxième Partie : Le jeu complet à deux joueurs

Pré-requis : Taper ceci, puis exécuter la fonction essai() pour comprendre son utilité.

```
def essai():
    k=0
    while k<4:
        lettre=input("Tapez une lettre : ")
        print("Vous avez entré la lettre",lettre)
        k=k+1
```

Les règles du jeu complet

Il y a toujours 18 bouts de bois, mais la composition du jeu change aléatoirement, il peut y avoir entre 0, 1, 2, .. ou plus de grands, de moyens, de petits.

Le joueur n°1 commence en mettant un bâton sur la table, le joueur n°2 pose à son tour u bâton sur la table avec les règles suivantes :

- le plus grand bâton gagne et rapporte le total des points correspondant aux deux bâtons.
- en cas de match nul, les deux bâtons sont écartés et personne ne marque de points
- le joueur qui gagne commence au prochain tour. En cas de match nul, c'est celui qu a joué en dernier qui recommence.

Exemple de déroulement d'une partie :

Le joueur 1 a obtenu [G,G,M,M,P,P,P,P,P] et le joueur 2 [G,M,M,M,M,P,P,P,P].

tour n°1	joueur 1 : G	joueur 2 : P	le joueur 1 gagne 4 points er rejoue.
tour n°2	joueur 1 : M	joueur 2 : P	le joueur 1 gagne 3 points et rejoue.
tour n°3	joueur 1 : M	joueur 2 : G	le joueur 2 gagne 5 points et rejoue.
tour n°4	joueur 2 : P	joueur 1 : G	le joueur 1 gagne 4 points et rejoue.
tour n°5	joueur 1 : P	joueur 2 : M	le joueur 2 gagne 3 points et rejoue.
tour n°6	joueur 2 : P	joueur 1 : P	match nul et c'est le joueur 1 qui rejoue.
tour n°7	joueur 1 : P	joueur 2 : M	le joueur 2 gagne 3 points et rejoue.
tour n°8	joueur 2 : M	joueur 1 : P	le joueur 2 gagne 3 points et rejoue.
tour n°9	joueur 2 : M	joueur 1 : P	le joueur 2 gagne 3 points et c'est fini.

Bilan : le joueur 2 a gagné 17 points à 11.

Question n°5 : Ecrire une fonction creationJeu() qui renvoie une liste de 18 bouts de bois composée de façon aléatoire entre les petits, les moyens et les grands.

Question n°6 : Ecrire une fonction deroulementJeu() qui affiche au début deux listes de bouts de bois, puis qui déroule le jeu en demandant au fur et à mesure à chaque joueur ce qu'il joue.

Le jeu affichera le vainqueur en fin de partie.

Exemple d'affichage dans la console :

```
>>> deroulementJeu()
 Liste du joueur 1 : [G,G,M,M,P,P,P,P,P], liste du joueur 2 [G,M,M,M,M,P,P,P,P]
 Le joueur 1 : G
 Le joueur 2 : P
 Le joueur 1 gagne 5 points
 Le joueur 1 : ...
```

Troisième Partie : un joueur contre l'ordinateur

Question n°7 : Ecrire une fonction moiContreMachine() qui affiche au début ma liste de bouts de bois et qui me fait jouer contre l'ordinateur. Dans cette question l'ordinateur joue au hasard et c'est lui qui joue en premier.

Question n°8 : Ecrire une fonction moiContreMachine2() qui me fait jouer contre l'ordinateur. Dans cette question l'ordinateur ne joue pas toujours au hasard.

Exemple :

- Quand il joue en premier, il joue au hasard.
- Quand il joue en second, il essaie de gagner et il ne gaspille pas inutilement ses bâtons, notamment :
    - si je joue un petit et qu'il dispose d'un moyen il ne va pas jouer un grand.
    - si je joue un grand et qu'il n'a plus de grand il va de préférence jouer un petit.