ITI 1520 Devoir 4

Instructions

Répondez à la question 1 dans le fichier d4q1.py, et la question 2 dans le fichier d4q2Lib.py.
Compressez le tout dans D4-NumeroEtudiant.zip.

N'oubliez pas d'ajouter des commentaires dans chaque programme pour expliquer le but du programme, la fonctionnalité de chaque fonction et le type de ses paramètres ainsi que le résultat, et les parties compliquées de vos programmes. Des points vous seront enlevés si les commentaires sont déficients.

Question 1 (8 points) Créez une fonction en Python appelée ajoute, qui prend une matrice et qui modifie la matrice en ajoutant 1 à tous les éléments. Créez une autre fonction Python appelée ajoute_V2, qui prend une matrice et qui retourne une nouvelle matrice contenant les valeurs de la matrice donnée comme paramètre incrémentée avec 1, sans la modifier.

Dans la partie principale du programme, demandez à l'usager d'introduire une matrice, invoquez la fonction ajoute et affichez la matrice modifiée. Ensuite, appelez la fonction ajoute_V2 avec la matrice comme paramètre, et affichez la matrice résultat. Affichez aussi la matrice initiale pour vérifier qu'elle n'est pas changée.

Les deux fonctions doivent convenir pour des matrices de toutes dimensions, même pour des listes qui ne sont pas des matrices (les listes intérieures ne sont pas nécessairement de la même taille).

Exemples :

```
Entrer les éléments de la matrice avec espaces entre les colonnes.
Une rangée par ligne, et une ligne vide à la fin.
1 2 3
4 5 6

La matrice est:
[[1, 2, 3], [4, 5, 6]]
Apres exécution de la fonction ajoute, la matrice est:
[[2, 3, 4], [5, 6, 7]]
Une nouvelle matrice créée avec ajoute_V2:
[[3, 4, 5], [6, 7, 8]]
Apres exécution de la fonction ajoute_V2, la matrice initiale est:
[[2, 3, 4], [5, 6, 7]]
```

Question 2 (12 points)
C'est le temps de jouer un jeu de X-O. Vous devez donc compléter le programme pour ce jeu.

Une partie de ce programme vous est donnée dans deux fichiers: d4q2.py (qui est complet, inutile de le modifier) et d4q2Lib.py (que vous devez compléter). L'annexe contient des exemples des messages affichés pendant le jeu.

1) Le programme principal contrôle le jeu.
a. Il demande à l'usager de commencer un jeu; si la réponse n'est pas o or O, le programme prend fin; Si la réponse est o or O, il dirige le jeu en utilisant les opérations suivantes:
   i. efface le tableau (avec la fonction effaceTableau(tab)).
   ii. affiche le tableau (avec la fonction afficheTableau(tab)).
   iii. joue une étape (avec la fonction joue(tab, joueur) – incluant la demande faite au joueur pour la nouvelle position).
   iv. vérifie si le joueur a gagné ou s'il y a match nul (avec la fonction verifieGagner(tab)).
   v. si le jeu n'est pas complet, joue encore une étape avec l'autre joueur (reprend avec iii).
Note: la matrice du tableau du jeu est créée dans la partie principale du programme en d4q2.py et passée comme référence aux autres fonctions.
b. après chaque jeu, il demande de commencer un autre jeu (reprend avec a.).
2) la fonction verifieGagner appelle les fonctions suivantes :
a. testLignes(tab) – pour vérifier si une ligne est gagnante.
b. testCols(tab) – pour vérifier si une colonne est gagnante.
c. testDiags(tab) – pour vérifier si une diagonale est gagnante.
d. testMatchNul(tab) – pour vérifier s''il y a match nul.

Votre travail consiste à compléter les fonctions suivantes qui sont dans le fichier d4q2Lib.py:
- effaceTableau (tab)
- verifieGagner(tab)
- testLignes(tab)
- testCols(tab)
- testDiags(tab)
- testMatchNul(tab)

Notes:

1) La fonction verifieGagner affiche le message "Match nul" au lieu de "Joueur X a gagné!" ou "Joueur O a gagné!" quand le match est nul.
2) La nouvelle ligne et la nouvelle colonne sont données avec input() et elles sont stockées dans une liste avec deux éléments (le premier est la ligne, et le deuxième la colonne).
3) Les fonctions testLignes, testCols, and testDiags, retournent un des caractères '-', 'X' or 'O'.
Si '-' a est retourné, personne n'a gagné, autrement le caractère pour le joueur gagnant est retourné.
4) La documentation pour chaque fonction est disponible dans le fichier d4q2Lib.py.

Optionnellement, pour des points supplémentaires (5 bonus points): développer une deuxième version de jeu X-O, avec une interface graphique pour afficher le tableau du jeu dans la partie principale de programme (et pour donner la prochaine position d'un X ou d'un O avec un clic de souris). Voir la documentation du module turttle ou tkinter.

Annexe – exemple jeu

```
Commencer un jeu (O ou N): O
  0 1 2
0 - - -
1 - - -
2 - - -
Joueur X, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 1
Colonne: 1
  0 1 2
0 - - -
1 - X -
2 - - -
Joueur O, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 0
Colonne: 0
  0 1 2
0 O - -
1 - X -
2 - - -
Joueur X, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 1
Colonne: 5
Joueur X, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 1
Colonne: 3
Joueur X, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 2
Colonne: 2
  0 1 2
0 O - -
1 - X -
2 - - X
Joueur O, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 0
Colonne: 0
La position 0 0 est occupée
Joueur O, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 0
Colonne: 2
  0 1 2
0 O - O
1 - X -
2 - - X
Joueur X, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 1
Colonne: 0
  0 1 2
0 O - O
1 X X -
2 - - X
Joueur O, SVP donner la ligne et la colonne de 0 à 2 :
Ligne: 0
```

```
Colonne: 1
Joueur O a gagné!
  0 1 2
0 O O O
1 X X -
2 - - X
Commencer un jeu (O ou N): N
```