

ITI 1520 - Devoir 4

Disponible: le 3 nov, 2024

Date de remise: mardi, le 19 nov, 2024, 22:00

SVP notez que le devoir n'est pas accepté après cette date.

Vous devez faire ce travail **en groupes de deux étudiants. Groupes d'un étudiant sont possible si quelqu'un a une préférence forte, mais les groupes de deux sont préférable.** Il y a des groupes BrightSpace pour vous inscrire vous-même.

Répondez à les question 1,2,3 dans le fichier d4.py, et à la question 4 dans les fichiers d4q4.py et jeuSudoku.py. Mettez tous les fichiers dans un répertoire compressé d4.zip pour soumission dans le campus virtuel.

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 (docstring), et les parties compliquées de vos programmes. Des points vous seront enlevés si les commentaires sont déficients.

Les noms des fichiers et les noms des fonctions doivent être les noms requis, parce qu'on va utiliser des tests automatiques pour la correction. Si le nom d'une fonction n'est pas le nom requis, la note sera zéro pour la fonction. Si vos fichiers .py donne erreur de syntaxe, la note sera zéro.

Si vous envoyez le devoir plusieurs fois, la dernière version sera note.

SVP noter qu'on va utiliser un outil logiciel pour détecter du plagiat. En cas deux devoirs sont identique ou très similaires, la note sera zéro pour les deux.

Le but de ce devoir est de tester les concepts de Module 7 et Module 8.

Ajouter une déclaration dans un fichier **declaration_VOTRE_NOM.txt**. avec votre soumission en BrightSpace (dans le fichier d4.zip)

Notes pour le fichier **declaration_VOTRE_NOM.txt** :

Ce fichier doit contenir la référence pour code qui n'est pas écrit par vous même **si c'est le cas**.

Ça inclue code donné par un collègue ou autre personne, ou trouvé sur l'internet, réseaux sociale (comme **Chegg, CourseHero, Stack Overflow, discord**, ou autre), ou généré par **ChatGPT** ou autre assistant virtuel. Ça n'inclue pas le code donne en BrightSpace dans les notes de classe, labo, etc.

Pour chaque question ou vous avez utilisé du code donné ou trouvé, il faut:

1. le numéro de la question
2. copier-coller le code emprunté ou généré. Ça inclue code donné/trouvé et modifier très peu.
3. le nom de la source: personne, site Internet ou autre

Vous pouvez perdre des points pour la question, mais au moins vous évitez une accusation de plagiat. En cas de plagiat, la note est zéro et le cas doit être rapporté au doyen.

Le même est valable si quelqu'un montre son code a un collègue.

Si vous n'utilisez pas cette déclaration, c'est équivalent à déclarer que tous le code a été écrit par vous-même.

Barème : total de **24** points. Devoir 4 est 4% de la note finale.

Part1.

Question 1. (4 points) Créez une fonction Python appelée **transl(d, s)** qui prends comme entrée un dictionnaire Python d et une chaîne des caractères s, et retourne la traduction de s en d. Si s est en anglais, la fonction retourne la traduction en français. Si s est en français, la fonction retourne la traduction en anglais. Si s n'est pas dans le dictionnaire ni en français ni en anglais, la fonction retourne la chaîne "Unknown". Voir des exemples des tests ci-dessous.

Ajouter le dictionnaire dans votre programme principal, comme une variable globale (bien sûr la fonction doit marcher pour n'importe quel dictionnaire similaire):

```
d = {"apple": "pomme", "banana": "banane", "pear": "poire", "plum": "prune"}
```

Question 2 (4 points). Créez une fonction en Python nommée **setOp(list1, list2)** qui prends comme entrée deux listes (nommée, par exemple, list1 et list2) et retourne un set avec tous les éléments des deux listes, sans répétitions.

Question 3. (4 points) Créez une fonction Python appelée **matrixMinMax(m)** qui prends une liste 2D, nommée m, par exemple, et retourne un tuple avec deux valeurs, le minimum et le maximum dans m. On fait l'hypothèse que la liste 2D n'est pas vide.

Testez les fonctions dans l'interpréteur Python ou avec le unit test fourni.

Voici quelques cas de test (testez aussi avec autres valeurs).

Optionnel : Ajouter un programme principal dans votre fichier d3.py pour appeler vos fonctions avec les valeurs ci-dessous.

```
#test Q1
transl(d, "pear")
'poire'
transl(d, "poire")
'pear'
transl(d, "apricot")
'Unknown'

#test Q2
setOp([1, 2, 2, 3], [2, -6, 8, 7])
{1, 2, 3, 7, 8, -6}
setOp([], [1, 4, 2, 4, 6])
{1, 2, 4, 6}
setOp([1, 1, 1, 1], [])
{1}
setOp([1], [2, 3, 2, 2])
{1, 2, 3}

#test Q3
matrixMinMax([[1, 5], [2, 8]])
(1, 8)
matrixMinMax([[1, 5, 10], [2, 8, -1]])
(-1, 10)
matrixMinMax([[2, 8, -1]])
(-1, 8)
matrixMinMax([[1], [1]])
(1, 1)
```

Part 2 (12 points) Jeu

Question 4. On va programmer le jeu Sudoku (<https://fr.wikipedia.org/wiki/Sudoku>). Vous devez donc compléter le programme pour ce jeu. Une partie de programme vous est donnée dans deux fichiers: **d4q3.py** et **jeuSudoku.py** (que vous devez compléter). Ne modifier pas les prototypes des fonctions. Vous devez implémenter les fonctions pour ce jeu. Un exemple d'exécution du jeu est donné à la fin de ce document.

Le programme principal contrôle le jeu (il est déjà donné).

Au début un menu de 3 choix est affiché à l'utilisateur. Ensuite, il demande à l'utilisateur d'entrée son choix. On suppose que l'utilisateur va entrer exactement le bon choix (un choix entre 1 et 3) donc la validité du choix n'est pas vérifiée.

Le choix 1 : il réinitialise la grille de Sudoku par la matrice donnée dans le programme principal (dans le fichier d4q4.py). Par la suite, le jeu commence en suivant les mêmes étapes que le choix 2.

Le choix 2 : le jeu commence. Il affiche la grille à l'écran et puis demande à l'utilisateur d'entrer la case à jouer (il doit entrer le numéro de la ligne, de la colonne et le nombre qu'il veut ajouter). Si la case à jouer est correcte (une case valide), alors il faut afficher le message Bravo !!. Sinon, le message Echec sera affiché. Si la case est déjà remplie ou si la grille de Sudoku est pleine, il affiche le message approprié à l'utilisateur.

Le choix 3 : Le jeu s'arrête. Le message Au revoir ! est affiché.

Afin de bien contrôler le jeu, il faut implémenter les fonctions suivantes :

afficherGrille() : permet d'afficher une grille Sudoku sous format d'une matrice 9x9. (cette fonction est déjà donnée)

verifierLigne() : permet de vérifier si le nombre à ajouter existe déjà sur la ligne.

verifierCol() : permet de vérifier si le nombre à ajouter existe déjà sur la colonne.

verifierSousGrille() : permet de vérifier si le nombre à ajouter existe déjà sur la sous grille où l'ajout sera fait.

verifierGagner() : vérifier si la grille est complète c.-à-d. il n'y a aucun case nul. Dans ce cas, la fonction retourne True.

verifierValide() : retourne True si la case est valide et False sinon (appelle les fonctions verifierLigne, verifierCol et verifierSousGrille) .

jouer() : ajouter une case dans la grille de Sudoku seulement si la case est valide.

Notez que:

1) Après chaque choix de menu, on réaffiche la grille après modification (s'il en y a) et le menu avec les options et redemande à l'utilisateur d'entrer de nouveau un entre choix jusqu'à que l'utilisateur choisis l'option 3 ou la grille est complétée.

2) La case à ajouter est demandé à l'utilisateur avec input(). On ne vérifie pas la validité de ces entrées (on suppose que l'utilisateur il va entrer toujours un nombre entre 1 et 9 et un numéro de ligne et numéro de colonne entre 0 et 8).

3) Les fonctions *verifierLigne()*, *verifierCol()*, *verifierSousGrille()* retournent *True* si le nombre n'existe pas sur la ligne ou la colonne ou la sous-grille.

Annexe – exemple jeu

Menu: 1- Commencer un nouveau jeu.
2- Continuer le jeu.
3- Quitter le jeu.

SVP entrez votre choix (1, 2 ou 3): 1

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 8 | 6 | 9 | 1 | 0 | 4 | 7 |
| 7 | 4 | 6 | 5 | 3 | 2 | 8 | 1 | 9 |
| 1 | 9 | 2 | 7 | 8 | 4 | 3 | 5 | 6 |
| 8 | 7 | 1 | 2 | 6 | 3 | 4 | 9 | 5 |
| 3 | 2 | 9 | 4 | 5 | 7 | 1 | 6 | 8 |
| 4 | 6 | 5 | 9 | 1 | 8 | 7 | 2 | 3 |
| 6 | 1 | 4 | 3 | 7 | 9 | 5 | 8 | 2 |
| 9 | 8 | 3 | 1 | 2 | 5 | 6 | 7 | 4 |
| 2 | 5 | 0 | 8 | 4 | 6 | 9 | 3 | 1 |

Entrez votre choix de case ligne: 0

Entrez votre choix de case col: 6

Entrez votre choix de case num: 2

Bravo!!

| | | | | | | | | |
|---|---|---|---|---|---|----------|---|---|
| 5 | 3 | 8 | 6 | 9 | 1 | 2 | 4 | 7 |
| 7 | 4 | 6 | 5 | 3 | 2 | 8 | 1 | 9 |
| 1 | 9 | 2 | 7 | 8 | 4 | 3 | 5 | 6 |
| 8 | 7 | 1 | 2 | 6 | 3 | 4 | 9 | 5 |
| 3 | 2 | 9 | 4 | 5 | 7 | 1 | 6 | 8 |
| 4 | 6 | 5 | 9 | 1 | 8 | 7 | 2 | 3 |
| 6 | 1 | 4 | 3 | 7 | 9 | 5 | 8 | 2 |
| 9 | 8 | 3 | 1 | 2 | 5 | 6 | 7 | 4 |
| 2 | 5 | 0 | 8 | 4 | 6 | 9 | 3 | 1 |

Menu: 1- Commencer un nouveau jeu.

2- Continuer le jeu.

3- Quitter le jeu.

SVP entrez votre choix (1, 2 ou 3): 2

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 8 | 6 | 9 | 1 | 2 | 4 | 7 |
| 7 | 4 | 6 | 5 | 3 | 2 | 8 | 1 | 9 |
| 1 | 9 | 2 | 7 | 8 | 4 | 3 | 5 | 6 |
| 8 | 7 | 1 | 2 | 6 | 3 | 4 | 9 | 5 |
| 3 | 2 | 9 | 4 | 5 | 7 | 1 | 6 | 8 |
| 4 | 6 | 5 | 9 | 1 | 8 | 7 | 2 | 3 |
| 6 | 1 | 4 | 3 | 7 | 9 | 5 | 8 | 2 |
| 9 | 8 | 3 | 1 | 2 | 5 | 6 | 7 | 4 |
| 2 | 5 | 0 | 8 | 4 | 6 | 9 | 3 | 1 |

Entrez votre choix de case ligne: 1

Entrez votre choix de case col: 2

Entrez votre choix de case num: 7

Case déjà remplie !!

Menu: 1- Commencer un nouveau jeu.

2- Continuer le jeu.

3- Quitter le jeu.

SVP entrez votre choix (1, 2 ou 3): 2

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 8 | 6 | 9 | 1 | 2 | 4 | 7 |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 4 | 6 | 5 | 3 | 2 | 8 | 1 | 9 |
| 1 | 9 | 2 | 7 | 8 | 4 | 3 | 5 | 6 |
| 8 | 7 | 1 | 2 | 6 | 3 | 4 | 9 | 5 |
| 3 | 2 | 9 | 4 | 5 | 7 | 1 | 6 | 8 |
| 4 | 6 | 5 | 9 | 1 | 8 | 7 | 2 | 3 |
| 6 | 1 | 4 | 3 | 7 | 9 | 5 | 8 | 2 |
| 9 | 8 | 3 | 1 | 2 | 5 | 6 | 7 | 4 |
| 2 | 5 | 0 | 8 | 4 | 6 | 9 | 3 | 1 |

Entrez votre choix de case ligne: 8

Entrez votre choix de case col: 2

Entrez votre choix de case num: 5

Echec!!

Menu: 1- Commencer un nouveau jeu.

2- Continuer le jeu.

3- Quitter le jeu.

SVP entrez votre choix (1, 2 ou 3): 2

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 8 | 6 | 9 | 1 | 2 | 4 | 7 |
| 7 | 4 | 6 | 5 | 3 | 2 | 8 | 1 | 9 |
| 1 | 9 | 2 | 7 | 8 | 4 | 3 | 5 | 6 |
| 8 | 7 | 1 | 2 | 6 | 3 | 4 | 9 | 5 |
| 3 | 2 | 9 | 4 | 5 | 7 | 1 | 6 | 8 |
| 4 | 6 | 5 | 9 | 1 | 8 | 7 | 2 | 3 |
| 6 | 1 | 4 | 3 | 7 | 9 | 5 | 8 | 2 |
| 9 | 8 | 3 | 1 | 2 | 5 | 6 | 7 | 4 |
| 2 | 5 | 0 | 8 | 4 | 6 | 9 | 3 | 1 |

Entrez votre choix de case ligne: 8

Entrez votre choix de case col: 2

Entrez votre choix de case num: 7

Bravo!!

| | | | | | | | | |
|---|---|----------|---|---|---|---|---|---|
| 5 | 3 | 8 | 6 | 9 | 1 | 2 | 4 | 7 |
| 7 | 4 | 6 | 5 | 3 | 2 | 8 | 1 | 9 |
| 1 | 9 | 2 | 7 | 8 | 4 | 3 | 5 | 6 |
| 8 | 7 | 1 | 2 | 6 | 3 | 4 | 9 | 5 |
| 3 | 2 | 9 | 4 | 5 | 7 | 1 | 6 | 8 |
| 4 | 6 | 5 | 9 | 1 | 8 | 7 | 2 | 3 |
| 6 | 1 | 4 | 3 | 7 | 9 | 5 | 8 | 2 |
| 9 | 8 | 3 | 1 | 2 | 5 | 6 | 7 | 4 |
| 2 | 5 | 7 | 8 | 4 | 6 | 9 | 3 | 1 |

Bravo!! Vous avez gagné