1 Modélisation de la grille de jeu.

1. Ouvrir Thonny et créer un fichier **NomSokoban.py**.
2. Définir la variable :

```
Grille= [[0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 1, 3, 2, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 1, 1, 2, 2, 3, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 1, 2, 2, 3, 2, 3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0],
[1, 1, 1, 2, 1, 2, 1, 1, 2, 1, 0, 0, 0, 1, 1, 1, 1, 1],
[1, 2, 2, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 4, 1],
[1, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1],
[1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 5, 1, 1, 2, 2, 2, 2, 1],
[0, 0, 0, 0, 1, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1],
[0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]]
```

Cette grille contient :
- des 0 : rien
- des 1 : mur
- des 2 : emplacement libre
- des 3 : caisse
- des 4 : case objectif

3. Définir les variables globales :

```
nb_caisse=6
ligneSokoban = 2
colonneSokoban= 5
```

4. Quel est le résultat de **Grille[ligneSokoban][colonneSokoban]** ?
5. Localiser le 5 dans la variable Grille.
6. Modifier les variables **ligneSokoban** et **colonneSokoban** pour que **Grille[ligneSokoban][colonneSokoban]** renvoi 5 (qui désigne le Sokoban).
7. On souhaite avoir un visuel plus "sympathique" du jeu :
   A partir de la variable **Grille**, les 0 seront remplacés par " ", les 1 par "M", les 2 par " ", les 3 par 'C', les 4 par "O" et le 5 par "S".
   Compléter la procédure ci-dessous qui permet d'afficher une ligne du plateau de jeu. Par exemple :

```
>>>representer([1,1,1,1,1,2,1,1,1,2,1,5,1,1,2,2,4,4,1])
MMMMM  MMM  MSMM    OOM
```

```
def representerLigne(ligne) :
    affichage=""
    ..................# Acompléter
        if case==0 :
            symbole=" "#rien
        elif case==1 :
            symbole="M"#mur
        elif case==2 :
            symbole=" "#libre
        elif case==3 :
            symbole="C"#caisse
        elif case==4 :
            symbole="O"#objectif
        elif case==5 :
            symbole="S"#sokoban
        affichage=affichage+symbole
    print(affichage)
```

8. Compléter la procédure pour avoir un visuel complet du jeux (utiliser la procédure **representerLigne**) :

```
def representer(Grille) :
```

```
>>> representer(Grille)

            M M M M M
            M         M
               M C       M
      M M M       C M M
      M       C     C     M
 M M M    M    M M    M          M M M M M M
 M        M    M M    M M M M M          O M
 M    C       C                          M
 M M M M M    M M M    M  S M M           M
            M                M M M M M M M M M
            M M M M M M M
```

2 Déplacement du Sokoban

9. Définir la procédure suivante :

```
def deplacer(direction) :
        global Grille
        global ligneSokoban
        global colonneSokoban
        global nb_caisse
```

La variable locale **direction** est une chaîne de caractère :
— "z" : aller en haut
— "q" : aller à gauche
— "d" : aller à droite
— "s" : aller en bas

2.1 "z" : monter le Sokoban

10. Compléter la procédure **deplacer** avec :

```
if direction=="Z" or direction=="z" :
```

2.1.1 Case libre

11. Au départ, la case au dessus du Sokoban est-elle une case libre (2 dans la **Grille**) ?

12. Compléter la procédure **deplacer** avec :

a. La commande **if Grille[ligneSokoban-1][colonneSokoban]==2**. Que permet de faire ce test ?

b. Affecter à la variable **Grille[ligneSokoban][colonneSokoban]** la valeur 2. (On met libre à la place du Sokoban)

c. On va monter le Sokoban, en ajoutant 1 à la variable **ligneSokoban**.

d. Affecter à la variable **Grille[ligneSokoban][colonneSokoban]** la valeur 5. (On met le Sokoban une case plus haut)

13. Tester la procédure avec :

```
>>> representer(Grille)
>>>deplacer("z")
>>> representer(Grille)
>>>deplacer("z")
>>> representer(Grille)
```

2.1.2 Case cadeau

14. Mettre en place un test qui permet de vérifier que la case au dessus du Sokoban est une caisse

15. Compléter les deux cas qui vont alors se présenter :

a. Si la case après le cadeau est libre alors on peut monter la caisse puis le Sokoban. Cette condition ce traduit en python par :

```
if Grille[ligneSokoban-2][colonneSokoban]==..... :
        Grille[ligneSokoban][colonneSokoban]=2 # On remplace le 5 par un 2
        ligneSokoban=..... # On monte le Sokoban
        Grille[ligneSokoban][colonneSokoban]=..... # On met le Sokoban
        Grille[.....][colonneSokoban]=3 #on met la caisse au dessus du Sokoban
```

b. Si la case après le cadeau est un objectif alors la caisse disparait et le Sokoban monte. Cette condition ce traduit en python par :

```
if Grille[ligneSokoban-2][colonneSokoban]==..... :
        Grille[ligneSokoban][colonneSokoban]=2 # On remplace le 5 par un 2
        ligneSokoban=..... # On monte le Sokoban
        Grille[ligneSokoban][colonneSokoban]=..... # On met le Sokoban
        nb_caisse=..... # On diminue le nb de caisse de 1.
```

16. Tester la procédure **deplacer**. (On modifiera à la main la variable **Grille** pour vérifier que Sokoban est capable de "monter" une caisse.)

2.2 "s" :descendre le Sokoban

17. A l'aide du paragraphe précédent, compléter la procédure pour gérer la descente du Sokoban.

```
if direction=="S" or direction=="s" :
```

2.3 "q" : décaler à gauche le Sokoban

C'est à vous ....

2.4 "d" : décaler à droite le Sokoban

C'est encore à vous ....

3 Jouer

Dans la procédure **jouer_Sokoban()** ci-dessous, on a mis une variable **reponse** qui est une chaîne de caractère saisie par l'utilisateur.

18. D'après le code ci-dessous, quelles sont les valeurs possibles pour la variable **reponse** ?

19. Dans le code ci-dessous, pour chacune des variables dire si elle est locale ou globale et en donner le type.

20. Dans quel**S** cas une partie de Sokoban s'arrête ?

21. Compléter alors le ***while*** de la procédure ci-dessous.

```
def jouer_Sokoban() :
    global Grille
    global ligneSokoban
    global colonneSokoban
    global nb_caisse

    representer(Grille)
    reponse="nImporteQuoi"
    while ..... :
        reponse=input("Votre déplacement (z/q/s/d) ou r pour recommencer ? ")
        deplacer(reponse)
        representer(Grille)


    if reponse=="r" :
        print("ce n'est pas grave, tu feras mieux la prochaine fois...")
    else :
        print("Bravo!!!")


    try_again=input("Voulez vous refaire une partie (y/n - y par défaut)")
    if try_again!="n" :
        jouer_Sokoban()
    else :
        print("A bientôt")
```

4 Améliorons le graphisme ...

22. Dans le même dossier que **NomSokoban.py**, télécharger le fichier **python sokoban_graphisme.py** et l'image **mario.gif**.

23. Ajouter à la ligne 1 de votre fichier NomSokoban.py :

```
from sokoban_graphisme import *
```

24. Dans la procedure **jouer_Sokoban()**, remplacer les trois lignes de la boucle ***while*** par :

```
print("Votre déplacement (z/q/s/d) ou r pour recommencer ? ")
afficher_jeu(Grille)
deplacer(reponse)
```

5 Vous pouvez enfin jouer !

25. Trouver une solution à ce jeu.

26. Construire une autre grille de jeu pour Sokoban.