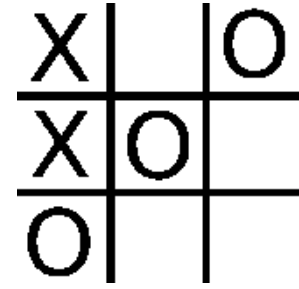



Le Morpion :

Règles du jeu : Le morpion est un jeu de réflexion se pratiquant à deux joueurs au tour par tour et dont le but est de créer le premier un alignement de ses pions sur une grille. Le premier joueur qui réalise cet alignement gagne un point. Une partie se joue en 5 points. Le joueur qui a perdu la partie précédente commence la partie suivante.



Quelques astuces avant de commencer :

Pour déterminer si un des deux joueurs a gagné la partie (*question g*) il est préférable de remplir la grille de jeu avec des 1 pour le "Joueur 1" et des -1 pour le "Joueur 2" c'est une astuce pour faciliter la détection d'un alignement de 3 cases. On initialise donc la grille avec des 0.

Pour que l'affichage de la grille du jeu dans la console soit plus lisible (*question f*) on associera aux pions de chaque joueur un symbole. Ce symbole est accessible en appuyant et maintenant la touche  tout en tapant le nombre associé à ce symbole. On donne ci-dessous les symboles possibles.

| | |
|--|---|
| ■ :  254 | ☺ :  1 |
| ○ :  9675 | ☹ :  2 |
| × :  0215 | |

a. **Proposer** une structure de données adaptée pour représenter les 9 emplacements de la grille de jeu.

b. **Proposer** une fonction "*fabric_grille*" qui renvoie la grille du jeu. Le résultat doit correspondre à :

```
>>> fabric_grille()
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
```

c. **Proposer** une fonction "*aff_grille*" qui prend en paramètre la grille précédente et l'affiche dans la console. Le résultat de l'appel de cette fonction doit correspondre à :

```
>>> aff_grille(fabric_grille())
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
```

d. **Proposer** une fonction "*saisie*" qui prend en paramètres le nom du joueur et la grille actuelle et attend que le joueur choisisse une case sur laquelle il souhaite mettre son pion. La saisie est de la forme lignecolonne. Seule les couples ('00','01','02','10','11','12','20','21','22') sont acceptés, tout autre choix doit conduire à réinterroger le joueur sur son choix. De plus on ne peut pas jouer sur une case déjà occupée. La fonction renvoie le choix de la case sous forme de tuple d'entiers.

```
>>> saisie("Joueur 1",grille)
Joueur 1 choisit sa case lignecolonne : 00 (str)
(0, 0) tuple (int, int)
>>> saisie("Joueur 2",grille)
Joueur 2 choisit sa case lignecolonne : 44
/!\ Entrée non valide
Joueur 2 choisit sa case lignecolonne : 00
/!\ Entrée non valide
Joueur 2 choisit sa case lignecolonne : 01
(0,1)
```

e. **Proposer** une fonction "*jeu*" qui remplisse et affiche la grille à l'aide des deux fonctions précédentes. Les actions à réaliser sont :

recupérer une grille construite

faire jouer alternativement les deux joueurs jusqu'à ce que la grille soit pleine* :

recupérer la case saisie par le joueur

lui attribuer 1 ou -1 suivant le joueur

afficher la grille

tester si la grille est pleine

* : on s'occupera plus tard de tester si un joueur à gagné.

Joueur 1 choisit sa case LigneColonne (0→2): 00

[1, 0, 0]

[0, 0, 0]

[0, 0, 0]

Joueur 2 choisit sa case LigneColonne (0→2): 11

[1, 0, 0]

[0, -1, 0]

[0, 0, 0]

Joueur 1 choisit sa case LigneColonne (0→2): 10

[1, 0, 0]

[1, -1, 0]

[0, 0, 0]

Joueur 2 choisit sa case LigneColonne (0→2): 20

[1, 0, 0]

[1, -1, 0]

[-1, 0, 0]

f. **Améliorer** votre fonction d'affichage pour qu'elle affiche votre grille par des chaînes de caractères de type : |espace"grille(LigneColonne)"espace|. Soit pour l'exemple précédent :

Joueur 1 choisit sa case LigneColonne (0→2): 00

| ■ | | |

| | | |

| | | |

Joueur 2 choisit sa case LigneColonne (0→2): 11

| ■ | | |

| | o | |

| | | |

Joueur 1 choisit sa case LigneColonne (0→2): 10

| ■ | | |

| ■ | o | |

| | | |

Joueur 2 choisit sa case LigneColonne (0→2): 20

| ■ | | |

| ■ | o | |

| o | | |

Pour tester si un joueur à gagné, il faut comptabiliser le nombre de 1 et de -1 présents sur les 3 lignes, les 3 colonnes et les 2 diagonales. Ce qui fait huit tests. Si la somme d'un de ces huit tests vaut 3 alors c'est le joueur1 qui a gagné, si la somme d'un de ces huit tests vaut -3 alors c'est le joueur2 qui a gagné, sinon la partie se poursuit.

- g. Proposer** une fonction "**test_grille**" qui prend en paramètre la grille actuelle, réalise les huit tests précédents et suivant les résultats de ces tests renvoie le nom du gagnant sinon None. Tester votre fonction avec :

```
assert test_grille([[1, 0, 0], [1, 0, 0], [1, 0, 0]]) == "Joueur 1"
assert test_grille([[-1, 0, 0], [-1, 0, 0], [-1, 0, 0]]) == "Joueur 2"
assert test_grille([[-1, 1, 1], [1, 1, -1], [-1, -1, 1]]) == None
assert test_grille([[-1, 1, 0], [1, 0, -1], [-1, -1, 1]]) == None
```

- h. Améliorer** l'affichage dans la console en affichant et numérotant les lignes et colonnes à partir de 1. Utiliser les caractères spéciaux pour afficher un les bords supérieur et inférieur.

