But

Le but est de réviser ce qui a été vu en classe de première, en codant un puissance 4.

La problématique

Le puissance 4 est un jeu qui se joue sur une grille 7×6 à deux joueurs. Les joueurs jouent l'un arpès l'autre et gagne si 4 de leur pions sont alignés.

Aide

◇ On rappelle que les commentaires triple quote `"""` en début de fonction sont affichés avec la commande `help()`.

```
def somme(*args: int) -> int:
    """
    renvoye la somme des paramètres
    ## exemples ##
    >>> somme()
    0
    >>> somme(1,4,7)
    12
    """
    return sum(args)

help(somme) # affiche l'ensemble du commentaire """
```

◇ Le format du texte est introduit par `\033[` suivi du code voulu.

```
print( "\033[31m" + "rouge" + "\033[0m" + "normal" )
```

Code par défaut 0m, code du texte souligné 4m.

Couleur du texte : noir 30m, rouge 31m, vert 32m, jaune 33m, bleu 34m, magenta 35m, cyan 36m, blanc 37m. (le code couleur du fond de texte est le même à partir de 40m)

Étapes

⚠ Pensez à commenter et tester vos fonctions (*cf.* aide).

a) télécharger le fichier `TNSI-puissance4-projet00.py` sur le site [forhan.maths.free.fr] et renommer-le.

b) La fonction d'affichage est donnée, compléter le code afin définir une grille 7×6.

c) Proposer une fonction `choixJoueur(jr: int) -> int`, qui affiche la grille et demande à l'utilisateur le numéro d'une colonne libre. Le numéro de colonne renvoyé sera nécessairement valide.

d) Commencer à écrire une procédure `main() -> None`, qui demande aux joueurs de jouer à tour de rôle et complète la grille en fonction de leur réponse jusqu'à ce que celle-ci soit pleine (on pourra diminuer `dimX` et `dimY` pour les tests).

e) La fonction `verifD(jr: int, x: int, y: int) -> bool` renvoie `True` si lorsque le joueur `jr` a posé un pion en `(x,y)` cela entraîne un alignement de ses pions selon la diagonale montante ou descendante, `False` sinon.

Proposer une fonction `verifH(jr: int, y: int) -> bool`, qui fasse la vérification pour un alignement horizontal d'un pion du joueur `jr` posé à la hauteur `y`.

f) De même avec la fonction `verifV(jr: int, x: int) -> bool` (V pour vertical).

g) Modifier la fonction `main()` afin que le jeu de puissance 4 soit fonctionnel.

h) Approfondissements :

- faire un menu qui demande le nom des joueurs et adapte les messages en fonction
- ajoute de la couleur pour visualiser les pions de chacun des joueurs et/ou le 4 aligné.
- proposer une IA

Rendu

À la fin du temps imparti, chaque groupe présentera en 2 minutes à l'oral les résultats qu'il a obtenu et déposera/enverra via l'ENT le fichier `TNSI-puissance4-NOM1-NOM2.py` contenant le code `python`.