Description

In this stage, we're going to analyze the game state to determine if either player has already won the game or it is still ongoing, if the game is a draw, or if the user has entered an impossible game state (two winners, or with one player having made too many moves).

Objectives

In this stage, your program should:

1. Take a string entered by the user and print the game grid as in the previous stage.
2. Analyze the game state and print the result. Possible states:

- `Game not finished` when neither side has three in a row but the grid still has empty cells.
- `Draw` when no side has a three in a row and the grid has no empty cells.
- `X wins` when the grid has three X's in a row (including diagonals).
- `O wins` when the grid has three O's in a row (including diagonals).
- `Impossible` when the grid has three X's in a row as well as three O's in a row, or there are a lot more X's than O's or vice versa (the difference should be 1 or 0; if the difference is 2 or more, then the game state is impossible).

In this stage, we will assume that either X or O can start the game.

You can choose whether to use a space or underscore `_` to print empty cells.

Note: List comprehensions in Python offer a concise way to create lists by iterating over sequences and applying conditions or transformations in a single line of code. They improve readability and efficiency compared to using traditional loops, making your code more Pythonic. For example, `[x ** 2 for x in range(10) if x % 2 == 0]` generates a list of squares for even numbers between 0 and 9.

Examples

The greater-than symbol followed by a space (`> `) represents the user input. Note that it's not part of the input.

Example 1:

```
> XXXOO__O_
---------
| X X X |
| O O _ |
```

```
| _ O _ |
---------
X wins
```

Example 2:

```
> XOXOXOXXO
---------
| X O X |
| O X O |
| X X O |
---------
X wins
```

Example 3:

```
> XOOOXOXXO
---------
| X O O |
| O X O |
| X X O |
---------
O wins
```

Example 4:

```
> XOXOOXXXO
---------
| X O X |
| O O X |
| X X O |
---------
Draw
```

Example 5:

```
> XO_OOX_X_
---------
| X O   |
| O O X |
|   X   |
```

```
---------
Game not finished
```

Example 6:

```
> XO_XO_XOX
---------
| X O _ |
| X O _ |
| X O X |
---------
Impossible
```

Example 7:

```
> _O_X__X_X
---------
|   O   |
| X     |
| X   X |
---------
Impossible
```

Example 8:

```
> _OOOO_X_X
---------
|   O O |
| O O   |
| X   X |
---------
Impossible
```