Assignment 4

Here you will continue to work towards building a tic-tac-toe game.

You will also begin to work with type-checking.

1. 45 points

Create an assign function as before that takes 3 arguments:
a position, a move, and a player,
and returns a new position with the player having made that move.

A position is represented as a tuple of tuples, as before.
e.g. the starting position is (('-','-','-'),('-','-','-'),('-','-','-'))

A move is a tuple consisting of a row, followed by a column.
For instance, (0,1) corresponds to placing in the top center.

The player is the character associated with a particular piece.
For instance, we could have the players use 'X', and 'O'.

Thus, finish the following definition:

```
def assign(position, move, player):
    #your code here
```

You do not have to worry about bounds checking, or checking if the square is empty first.

2. 45 points

First, let us create some global constants

```
first_player_wins = "first player wins"
second_player_wins = "second player wins"
tie = "tie"
ongoing = "ongoing"
```

Now define the function detect_result,
to take one argument, the position,
and return the correct result.

Thus, complete the following:

```
def detect_result(position):
    #your code here
```

Make sure that when you return the result, you use the constants above.

That is, you should return first_player_wins, second_player_wins, tie, or ongoing.

Assume that a position is represented as a tuple of tuples, as before.

3. 10 points

Install mypy-lang,
and write any program or function that has type annotations which you can check with mypy,
before running your program.