Description: The *prisoner’s dilemma* is a two-player game that is similar to every episode of Law & Order. Two criminal accomplices are apprehended by police. They are held and interrogated separately with no way to communicate with each other. The police don’t have enough evidence to charge the accomplices with a crime unless at least one of them confesses. Thus, each player must make a decision. They can confess to the crime and implicate their partner (this move is often called defect) or they can remain silent (often called cooperate). The sentence that each player receives is dependent not only on their own decision but on that of their accomplice. The graphic above shows the sentences for each player in each of the four possible outcomes.

So it seems like the best strategy is to simply keep quiet. However, a player receives better treatment (a reduced sentence or an increased reward, depending on the version of the game) if they confess when their partner remains silent. So each is left to wonder whether their accomplice is talking.

In the version of the game you will implement, the players will receive rewards rather than punishments. The matrix of rewards appears below.
The *iterated prisoner’s dilemma* game consists of a sequence of some number of rounds of the prisoner’s dilemma. Each round of IPD is a game of PD. The number of rounds is unknown to the players. A player’s score in an IPD game is the sum of the scores of in all of the rounds.

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>3 : 3</td>
<td>0 : 5</td>
<td></td>
</tr>
<tr>
<td>Defect</td>
<td>5 : 0</td>
<td>1 : 1</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Scoring for the version of IPD in this work. Note that these values represent rewards rather than penalties. Thus, a higher score is better.

Details: For this program, you will implement an iterated prisoner’s dilemma game. The number of rounds in the IPD game should be an optional parameter with a default value of 150. Player 1 in the game will use a strategy known as Tit-for-Tat. This strategy is simple but formidable. Tit-for-Tat cooperates in the first round 0. In every subsequent round $i$, it chooses whatever its opponent chose in round $i - 1$.

For Player 2, you will implement at least three other strategies. Your goal is to find a strategy that will outperform Tit-for-Tat in terms of total score (higher is better in our version of the game). Your strategies can “remember” the results of as many previous rounds as you like.

Tournament: If there is interest, we will hold a tournament to determine who created the best Tit-for-Tat killer. The winner will receive the only prize that matters: the undying admiration of their peers (you knew that was coming).

Your code: You should decompose your solution into a number of functions. Make use of optional parameters, multiple returns, etc. as needed. In addition, you should format your code using `__main__` and `main()` as we have seen in class. Comment your code and adhere to common coding conventions. If you use camel case naming, I’ll know that you are trying to make me cry.

Include a comment at the top that looks like this:

```python
#
# CS 224 Spring 2020 - After the Zombie Apocalypse
# Programming Assignment 4
#
# Your wonderful, pithy, erudite description of the assignment.
#
# Author: Your name here
# Date: April 8, 2020
#```
Submission: The name of your program must be `ipd.py`. Submit your solution as a compressed directory by 11:59 PM on the due date.