HW 5 Tic Tac Toe final version

Start Assignment

- Due Monday by 6pm
- Points 10
- Submitting a file upload
- File Types py
- **Don't forget to add your lastname_firstname_HW5 as the filename**
- Continue your tic tac toe program. **BOLD is what should be added**
  - Put up a menu
  - (1) Play tic tac toe
  - **(2) Human vs Computer.**
    - **Random decision on who goes first (1 point)**
    - **Computer goes in a random empty spot (1 point)**
  - **(3) Computer vs Computer (simulate 10,000 times)**
    - **Print out # of times X wins, # of times O wins, # of times cat game (1 point)**
    - **Who wins (first player or second)?**
      - **Keep track of first 2 moves. What combination is best for second player?**
        - **Print out 9 times 8 = 72 combinations and how often each computer one (2 point)**
          - **HELP: this is tough. I did a list of dictionaries. Here is how I initialized it**
            - ```
              history = [
                  {"first": "", "second": "", "results": {"xwins": 1, "owins": 1}}
              ]
              ```
          - **Then if you find the combination in the history list, update the xwins or owins**
          - **if you can not find the combination, create a new list element and update the xwins or owins (1 point)**
          - **Here is my function that does that:**
          - ```
            def record_game(current_game, history):
                found = False
                for final_result in history:
                    if final_result["first"] == current_game[0] and final_result["secon
            d"] == current_game[1]:
                        found = True
                        if current_game[2] == 'X':
                            final_result["results"]["xwins"] += 1
                        elif current_game[2] == 'O':
                            final_result["results"]["owins"] += 1

                if not found:
                    history.append({"first": current_game[0], "second": current_game[1],
            "results": {"xwins": 0, "owins": 0}})
            ```

```
        if current_game[2] == 'X':
            history[-1]["results"]["xwins"] += 1
        elif current_game[2] == 'O':
            history[-1]["results"]["owins"] += 1
```

- **(4) Computer vs Smarter computer**
  - **Put second player with extra logic**
    - **If second player can win in the next move, have it win. (1 points)**
    - **If second player sees player 1 will win in the next move, block it (1 point)**
  - **Make player 2 smarter. If you can win, then win; If you can block, then block**
  - **Add logic from above for best move for player 2. If 2nd player (1 point)**
  - **Print out # of times X wins, # of times O wins, # of times cat game (1 point)_**
- (4) exit game
- Write a function that prints_current_board. (2-D list or a 1-D list)
  - A board is a 3 by 3 with numbers 1 through 9 if empty and a "X" and "O" if taken
- Write a function that checks_if_valid
  - If valid replace the number with X or O; else try again
- Write a function that checks_if_won or if a cat game
  - ChatGPT has code we have not learned…use a set of "if statements" and "or"