CS 441 – Spring 2024 Program 1

This is an introductory program to get you familiar with programming in Racket. It's pretty much data in, data out.

Bowling is scored as follows:

- A game consists of 10 frames.
- In each frame, a bowler can roll up to 2 balls, in an attempt to knock down all 10 pins. Knocking down all the pins in one roll is a *strike.* Knocking down all the pins in 2 rolls is a *spare.* One or more pins left standing is a *blow,* or an *open* frame.
  - A blow scores the number of pins knocked down.
  - A spare scores 10, *plus* the number of pins knocked down on the bowler's next roll.
  - A strike scores 10, *plus* the number of pins knocked down on the bowler's next 2 rolls.
  - If a player makes a spare in the 10^{th} frame, they roll 1 extra ball to complete scoring for the spare. (They don't keep rolling if it's a strike; they roll one ball, and that's it.)
  - If a player makes a strike in the 10^{th} frame, they roll 2 extra balls; again, they don't keep rolling if the 2 balls are strikes or a spare.
  - The highest possible score (every roll a strike) is thus 300.

Scores are recorded as follows:

- For each frame, an X indicates a strike.
- A number followed by / indicates a spare.
- 2 numbers indicates a blow. The numbers are the number of pins knocked down by each ball.
- Any additional rolls needed after frame 10 are appended to the score.

Sample score:
A. Player 7 / X 5 4 X X 7 / 5 4 8 / X 8 / X

Frame 1: A spare; 10 + next roll = 10 + 10 = 20
Frame 2: A strike: 10 + next 2 rolls = 10 + 9 = 19 (Note that the score won't be known until the end of Frame 3)
Frame 3: An open frame: 9
Frame 4: A strike: 10 + next 2 rolls = 10 + 10 + 7 = 27
Frame 5: A strike: 10 + next 2 rolls = 10 + 10 = 20
Frame 6: A spare: 10 + next roll = 10 + 5 = 15
Frame 7: An open frame: 9
Frame 8: A spare: 10 + next roll = 10 + 10 = 20
Frame 9: A strike: 10 + next 2 rolls = 10 + 10 = 20
Frame 10: A spare: 10 + next roll = 20
(The last X is the extra ball needed to complete scoring for the spare in frame 10. It does not trigger additional rolls.)

Total score: 20 + 19 + 9 + 27 + 20 + 15 + 9 + 20 + 20 + 20 = 179

A team consists of 5 bowlers. A match is conducted with each bowler playing 3 games. The team with the higher total score wins the match. There is also a prize for the bowler with the highest total score for the 3 games.

You are given a data file, scores.txt. Put the data file in the same folder as your Racket source code. You may hard-code the name of the file, or ask the user for it. The file format is as follows:

- Name of the first team (1 word), on a line by itself.
- 15 games, each formatted onto a single line. The line begins with the first and last name of each player for that game, followed by frame scores separated by whitespace, as in the example above. There will be 5 players, and 3 games per player; however, the games will not be listed in any particular order.
- Name of the other team, followed by the 15 games for that team, in the same format as above.

Your program should report:
For each team:
    For each player:
        The 3 game scores, and the total score for that player
    The total score for that team

The winning team
The highest-scoring player. (If there is a tie, list all players with that score.)

Programming Notes:

- Think recursively!
- When decomposing the problem, design your functions with the idea that the output of any function is probably going to be the input to some other function.
- Use of ChatGPT and similar tools are allowed, but save your prompts. (And you'll probably find that it's useful for some things, but not others.)