Alten Canada - Python Technical Assignment

Your company organises volley tournaments all over the country, and they rely on a very old Python script someone wrote long ago. The code still works, but clearly has issues. They just hired you as a Python developer and you are going to work on everything related to how they calculate quotes that the teams have to pay for registering to a competition.

The way the process works is extremely manual and very error prone at the moment. Your company's staff takes care of having email conversations with the players, and updates csv files manually. This is clearly not ideal and will eventually lead to problems sooner or later.

The way they keep track of whether someone has already issued their payment, is by manually updating the *Paid* column in the *players.csv* file (more about it below). The only automated process (that's what the script is for) is to calculate the quote that each team still has to pay for participating in an event. This process usually happens at the end of the tournaments and takes into consideration many factors (described below) as well as whether someone has already paid their quote or not.

- The company has 2 main csv files they use to keep track of registrations and payments:

    - players.csv: this file keeps a list of the players, which tournament(s) they took or will take part to, their skill level and whether their quote has already been paid or not

    - tournaments.csv: this file keeps a list of the tournaments and their structure (number of games)

- In order to keep the tournaments accessible to people with any skill level, the company thought of building a payment system based on the skill level of each player. The better the skill level each participant has, the more they will pay to take part in an event. On the other hand, the lower the volleyball level is for a participant, the less they will pay to play.

  This is what the company came up with as a solution to have participants of all levels, as well as encouraging more people to take part in the events. The skill level of each participant is assessed by the members of the staff.

    - If someone has never participated in a tournament, they are assigned a skill level of 3 for the first tournament they take part in. In these cases, the csv contains a blank space for the skill level, not the value *3*.

- After the first tournament, each player has an assigned skill level, eventually adjusted manually by the organisers, were that level to change over time.

- The registration cost for each player is determined as follows:

    - Skill level 1: 1.5 $ per game
    - Skill level 2: 3 $ per game
    - Skill level 3: 5 $ per game
    - Skill level 4: 7 $ per game
    - Skill level 5: 10 $ per game

- For each game they are potentially going to play in a tournament (that is, were they to reach the final), they pay:

    - 1 time the cost of their skill level for each round-robin game
    - 2 times the cost for their skill level for each game of the elimination phase

There are 3 types of tournaments:

- Round-robin + round of 4:
    - 2 groups of 4 teams each
    - 3 group games
    - Semi-final
    - Final

- Round-robin + round of 8:
    - 4 groups of 4 teams
    - 3 group games
    - Round of 8 (quarter-final)
    - Semi-final
    - Final

- Round of 16:
    - No group games
    - Round of 16
    - Round of 8
    - Semi-final
    - Final

The cost is calculated based on the maximum potential number of games a team could play in a tournament, unregarding of how a team finishes in the competition. (e.g. if a team plays in a Round of 16 tournament, they would pay for 4 games even if they lose the first one, to cover for the costs of the organization and the final prize)

Lately, one of the organizers has decided to take part in a tournament, and registered her team to participate in a competition. Being an organizer, she has access to the files with the data, and since she does not know how to run the python script, she accesses the files and calculates the total for her team, preparing the money and putting them aside for the payment later on.

When the organization asks her to pay for the tournament registration, she finds out it seems like she didn't gather enough money for the total that is presented to her. She calculates the total again from the csv file and tells you about the difference between the two amounts (the manual one and the one calculated by the script). You conclude that the cause of the miscalculation is probably due to a bug in python script.

Your assignment is to:

1. Understand the scenario described above (not just the bug, the whole organizational flow) and analyse the code. Find the bug that caused the girl to have to pay an amount which is greater than the total calculated by her manually, and fix it.

   For point 1, save the script as main_1.py, and move to point 2, starting from a fresh copy of main.py if you wish to. **(Hint: it is a very small bug, and it can be fixed with probably even 1 line of code)**

2. Refactor the code completely in order to make it:

   a. **More understandable** (e.g. adding comments, explaining what the functions do (inside the code, with comments, without comments, with good variable names, in a README file, anything you can come up with)).
   The only limit you have for this, is that the two csv files need to have the exact same structure, in terms of columns, i.e. you can't change the structure for them, but you can add data if you wish to (especially for point 2b).

   b. **Less ad-hoc** for volleyball tournaments only. Your company plans to start organizing tournaments for other sports (soccer and tennis). They want an easy way of adding as many sports they can organize tournaments for in the future, and they decided to go ahead with this by using the column Sport in the tournaments.csv (which is already there with the value *Volleyball* for all rows).

      NB: as much as this is not the ideal technological path for the company to be able to add other sports to their schedule, this is the solution they came up with, and they won't change it (for now). So, for point 2b, it is a requirement for you not to change the structure of the files and just work with the python code as the only technology to implement the new behavior.

    c. **Tested**. The code needs to be tested and safe against exceptions that could occur during the calculation. Tests need to be formal Python tests, and you can use whichever framework or python method you prefer in order to achieve this.

3. After some months you worked for the company, you gained their trust, they liked the job you did on the script, and now they are asking you to not only re-think the Python part of the process, but to refactor the whole flow as opposed to the calculation part only.

   NB: there is no need for you to implement anything for this part specifically. This point is just about you coming up with ideas and ways of potentially implementing a system to improve the company process. For example, if one of the solutions you think of is to create a database containing tables for the entities of the process (players, tournaments, etc.), you don't need to actually create the database nor the python connectors to interact with those components, because it would most likely take too much time. Point 3 is part of the assignment to evaluate the capacity of the candidate to go beyond the technology used and come up with ideas that can be useful to the business as a whole. The ideas will be briefly discussed during the technical interview, but will definitely not be the main topic of it.