Programming Project / C++ – Table Tennis Match Management System–

Student 1: Ibrahim Rania

Student 2: Frasincar Raul

I. Task Description

Student 1 is responsible for the internal data management of the system:

- Managing the list of players and matches.
- Handling match result updates and computing win/loss statistics.
- Reading from and writing to persistent files (`players.csv`, `matches.csv`).

Student 2 is responsible for user interaction and command line argument interface:

- Designing how commands are input through the terminal.
- Parsing and validating user input format.
- Displaying formatted output for history and leaderboard.

II. Data Structures Used by the Team

The system uses the following classes:

- **`Player`**
  - Attributes: `std::string name`, `int wins`, `int losses`
- **`Match`**
  - Attributes: `std::string p1name`, `std::string p2name`, `int p1score`, `int p2score`

- **`MatchManager`**
    - Composition of `std::vector<Player>` and `std::vector<Match>`
    - Handles:
        - Data loading/saving from CSV files
        - Command parsing
        - Adding matches
        - Displaying history and leaderboard

III. File Structure

The following files will be used:

- **`players.csv`**
  Format: `<PlayerName>,<Wins>,<Losses>`

- **`matches.csv`**
  Format: `<Player1>,<Player2>,<Score1>,<Score2>`

These files are automatically read and updated by the program.

IV. Interacting with Executables

The application will offer the following options:

Add a match result:

```
/app add_match "Player1 21/17 Player2"
```

This will:

- Record a match with Player1 scoring 21, and Player2 scoring 17.
- Update the win/loss records.
- Save it to `matches.csv` and update `players.csv`.

Display the leaderboard:

```
/app leaderboard
```

This will print a list of all players sorted by:

1. Most wins
2. Fewest losses (if wins are tied)

View a player's match history:

```
/app history Player1
```

This will print:

- Every match involving Player1
- Each match's score and opponent
- Total wins and losses