Assignment Specifications

Problem: Data Processing

An acquaintance of yours who is a soccer fan, has been collecting information about the World Cup and stored this information in a set of 3 text-formatted files. 32 countries have qualified for the World Cup in November 2022 in Qatar. The information collected only concerns the group stage of the world cup when the 32 countries were divided based on a draw into 8 groups from A to H. The teams faced each other in their respective groups and the 2 best in each group advanced to what is known as the knockout stage where the round of 16, the quarter-finals, the semi-finals, and the finals were played. Your acquaintance knows that you are taking CMPUT 175 and therefore knows how to program in Python. They asked you to help write a program to process the data in these files and then format some specific information. Your Python program would use these text-formatted files as input and produce output lists that can be printed.

The files that you have as input are as follows:

The first file, **WC22Footballers.txt**, contains the list of all the players from all national teams who qualified for the World Cup. Each team has 26 players. Each line of the file is for one player and the country, the number assigned to the player, his position, his name, birthday and age. Since the names are international, they can have special characters. Therefore, the file is not in simple ascii but uses a codec encoding utf-8 to allow these special characters. Lines would look like the following:

```
France 25;MF;Eduardo Camavinga;10 November 2002 (aged 20)
France 26;FW;Marcus Thuram;6 August 1997 (aged 25)
Tunisia 1;GK;Aymen Mathlouthi;14 September 1984 (aged 38)
Tunisia 2;DF;Bilel Ifa;9 March 1990 (aged 32)
```

Each team has 26 players. There is a line in **WC22Footballers.txt** for each of the 26 players in the 32 different national teams. Each line contains several several pieces of information, organized as follows:

- Country name and t-shirt number. Each player in a team is assigned a unique number between 1 and 26.
- Player field position, where GF=Goalkeeper; DF=Defender; MF=Midfielder; and FW=Forward.
- Player first and last name
- Player date of birth followed by their age between parenthesis.

The fields described in the bullet points above are separated by semicolons (;). Please note that this file contains names of countries and people from all around the world. Thus it may contain special characters and is encoded in UTF-8.

The second file, **WC22GroupMatches.txt**, contains the list of all the matches played during the group stage as well as the scores and who scored the goals. Each line in the file concerns one match and has 5 fields separated by a semicolon (;). Examples from this file look like this:

```
F;Morocco;Canada;(7,19)(X5);1 December 2022
F;Croatia;Canada;(9,14,9,7)(19);27 November 2022
F;Croatia;Belgium;()();1 December 2022
F;Belgium;Canada;(23)();23 November 2022
G;Brazil;Switzerland;(5)();28 November 2022
G;Brazil;Cameroon;()(10);2 December 2022
```

In more details, file contains per line the following fields, separated by semicolons:

- World cup group of the match (A-H), the 8 groups in the World Cup.;
- Name of the first team;
- Name of the second team;
- Two lists of players who scored during the match. The lists are stored between parenthesis.
    - The first list contains the t-shirt numbers of the players who scored for the first team
    - The second list contains the t-shirt numbers of the players who scored for the second team
    - Players can score AGAINST their own team, that is they score a goal for the opposing team. In that case the t-shirt number of the player is listed in the opposing team list preceded by a letter 'X'
    - Empty lists mean that no player of the corresponding team scored in that game
- The date of the match in the format 'DD FULL_MONTH YYYY'

For example, the last line in the example above has ()(10). This means the first team Brazil did not score any goals, therefore the list is empty (). However, Cameroon scored one goal. The scorer was number 10 (i.e. Cameroon 10), that is Vincent Aboubakar from the previous file. That means the number in the list is the number of the player from that country. However, sometimes a player may accidentally score in his own team goal. This is called a self-goal. Players who score a self-goal are prefixed with an X. For instance, in the first line of the example above, the match Morocco against Canada in group F ended 2 against 1. Two goals for Morocco by 7 and 19 (i.e. Hakim Ziyech and Youssef En-Nesyri) but the Moroccan 5 (i.e. Nayef Aguerd) marked for Canada, against his own team (this is why 5 is preceded by the letter 'X').

The third file, **WC22YellowCards.txt**, contains the history of all yellow cards. Yellow cards are doled out to a player by the match referee when the player fouls another in a way deemed somewhat excessive or otherwise disrupts the game. There are also red cards. A red card also doled out by the referee means that a player is deemed to have committed a foul so egregious that they must be expelled from the game. So in this file, each line is for a doled out card. The line has many fields again separated by a semicolon (;). The first field is simply the two countries facing each other. The second field is the country of the player receiving the card. The third field is the name of the player receiving the card. The fourth field is the type of card: Y for Yellow and R for Red. Finally ,the last field is the time at which the card was given.

```
Belgium-Canada;Belgium;Yannick CARRASCO;Y;9
```

```
Belgium-Canada;Belgium;Thomas MEUNIER;Y;54
Belgium-Canada;Belgium;Amadou ONANA;Y;56
Belgium-Canada;Canada;Alphonso DAVIES;Y;81
Belgium-Canada;Canada;Alistair JOHNSTON;Y;83
```

For example, in the match Belgium versus Canada, Alphonso Davies from Canada received a Yellow card in the 81st minute.

For all input files, you can assume that there will not be whitespace at the beginning of each line, nor will there be blank lines in the file. You can also assume that all players have a unique country and number.

Your Python program must generate text files for the output and display the answer on the screen for the following questions:

1- List of countries per Group (sorted by country).

2- List of countries that pass to the knockout stage with the points they accumulated.

3- Provide the average age per country (sorted by country).

4- Display the histogram of player ages.

5- From each group, which country scored the most goals? Also provide the list of the players that scored for that country? **(optional question)**

6- Which player scored the most goals?

7- Which match had the most goals? **(optional question)**

8- Which match had the most yellow cards?

Note that to answer these questions, you should not read the three input files more than once. The files should be read once and the information stored in memory in appropriate data structures that would allow to answer the questions.

Expected output

1- The list should be displayed to the console as well as written on a text file **groups.txt**. The countries in each group should be sorted. An excerpt of the output should look like:

```
Group A
Ecuador
Netherlands
Qatar
Senegal
```

```
Group B
England
Iran
USA
Wales

Group C
Argentina
Mexico
Poland
Saudi Arabia

...
```

2- The list should be displayed to the console as well as written on a text file **knockout.txt**. The countries should be ranked alphabetically each line with the country name and the number of points. Only the top two countries per group pass to the knockout stage. The points are calculated based on the number of wins, draws and losses (Win=3 pts, draw=1 pt, loss=0 pt). If two countries in the same group sum up to the same number of points, use the goal difference in the group match between these two countries to distinguish between them. If they had a draw during that match then the number of yellow and red cards will demarcate them. The fewer yellow cards the better. A red card is equivalent to 4 yellow cards. Here is an example showing the top of the list:

```
Argentina    6 pts
Australia    6 pts
Brazil       6 pts
Croatia      5 pts
England      7 pts
France       6 pts
...
```

Notice that the country name takes 12 spaces. If it is less then padding is required to make sure the points are displayed in a straight column.

3- The list should be displayed to the console as well as written on a text file **ages.txt**. The average of the 26 players in each team needs to be calculated in addition to the total average for all players in the group stage. Here is an example output from the bottom of the list.

```
...
Spain        25.35 years
Switzerland  27.00 years
Tunisia      27.92 years
Uruguay      27.81 years
USA          25.15 years
Wales        26.38 years

Average Overall 26.93 years
```

Notice that the average age is with two digits after the decimal point and the country name takes 12 spaces. The country name is to be padded if necessary as before.

4- The histogram should be displayed to the console as well as written on a text file **histogram.txt**. The histogram should be formatted as seen below using stars (*) to draw the bars each star is equivalent to 5. That means that for instance 40 would be represented by 8 stars. To do so, divide the number by 5 and use the built-in round() function in Python. if the result is 0 (i.e. number below 5) use one star. here is what the histogram could look like:

```
18 years ( 6)*
19 years (13)**
20 years (28)******
```

```
21 years (35)*******
22 years (51)**********
23 years (54)***********
24 years (74)***************
25 years (83)*****************
26 years (61)************
27 years (70)**************
28 years (53)***********
29 years (63)*************
30 years (72)**************
31 years (41)********
32 years (41)********
33 years (28)******
34 years (19)****
35 years (16)***
36 years ( 9)**
37 years ( 6)*
38 years ( 2)*
39 years ( 5)*
40 years ( 1)*
```

5- **(optional)** The list should be displayed as well as written on a text file **maxgoals.txt**. For each of the 8 groups, we need to know the country that scored the highest number of goals during the group matches, along with the total number of goals. The names of the players that scored these goals and the number of goals each scored. If two countries (or more) in the same group total the same highest number of goals, they all need to be indicated. Here is an excerpt of possible output.

```
...
Group E- Spain 9 goals
Álvaro Morata 3
Carlos Soler 1
Dani Olmo 1
Gavi 1
Marco Asensio 1
Pau Torres 2

Group F - Croatia 4 goals
Andrej Kramarić 2
Lovro Majer 1
Marko Livaja 1

Group F - Morocco 4 goals
Hakim Ziyech 1
Romain Saïss 1
Youssef En-Nesyri 1
Zakaria Aboukhlal 1
...
```

6- The table should be displayed as well as written on a text file **scorer.txt**. Stick to the formatting. If more players are tied for the highest number of goals scored, all their names should be indicated. The table should look like this:

```
+-------+-------------+-------------------------+
|3 goals| Netherlands |  9 Cody Gakpo           |
|3 goals| England     | 11 Marcus Rashford      |
|3 goals| France      | 10 Kylian Mbappé        |
|3 goals| Spain       |  7 Álvaro Morata        |
+-------+-------------+-------------------------+
```

7- **(optional)** The output should be displayed as well as written on a text file **mostgoals.txt**. The date of the match should be output followed by the countries facing each other and the final score.

Finally the scorers should be listed. A player should be displayed as many times as the number of goals he marked. Here is a possible output:

```
The match with the most goals
21 November 2202 - Group B
England vs. Iran 6-2
England 22: Jude Bellingham
England 17: Bukayo Saka
England 17: Bukayo Saka
England 10: Raheem Sterling
England 11: Marcus Rashford
England 7: Jack Grealish
Iran 9: Mehdi Taremi
Iran 9: Mehdi Taremi
```

8- The output should be displayed as well as written on a text file **yellow.txt**. The output consists of the countries facing each other during the match, followed by the number of yellow cards each country received. The output could look like this:

```
Serbia vs Switzerland
Serbia: 7 YC
Switzerland: 4 YC
```

Note that the example output given above may not exactly be related to the input examples provided here. Indeed, the output depends on 3 input files. The input and sample output provided here are just examples to indicate the format of all of the files. **DO NOT hard code your output.** The TAs will evaluate your assignment with different input files.

Stick to the required output format. Do not try to make it "nicer" or "more practical", or anything else. This is what the customer wants.

General Guidelines

In addition to making sure that your code runs properly, we will also check that you follow good programming practices. For example, divide the problem into smaller sub-problems, and write functions to solve those sub-problems so that each function has a single purpose; use the most appropriate data structures for your algorithm; use concise but descriptive variable names; define constants instead of hardcoding literal values throughout your code; include meaningful comments to document your code, as well as docstrings for all functions; and be sure to acknowledge any collaborators/references in a header comment at the top of your Python file.

Restrictions for this assignment: you cannot use break/continue, and you cannot import any modules a part from the codecs module if you have to. Importing other modules will result in deductions.