PROGRAMMING CHALLENGE 15: FOOTBALL LEAGUE

An application is to be created to store a Football League table data. The total number of teams in the league is 10. The team names are stored in the file `TEAMS.txt`.

The results of the football matches are provided in file `RESULTS.txt`. Each match data takes up one line, for example:

```
MadUnited 2 Chelsand 1
```

That is, Mad United won Chelsand, scoring two goals and conceding one goal, or Chelsand lost to Mad United, scoring one goal and conceding two goals.

The League table that needs to be created has the following information:

Team	**P**	**W**	**D**	**L**	**GF**	**GA**	**GD**	**Points**
MadUnited	3	2	1	0	5	2	3	7
Chelsand	3	2	1	0	4	2	2	7

...

Legend:
P – games played
W – games won
D – games drawn
L – games lost
GF – goals for (scored against opponents)
GA – goals against (goals conceded by team)
GD – goal difference, i.e. GD = GF – GA
Points – computed based on 3 points per win, 1 point per draw and zero points per loss

Task 1

Write program code for a procedure `CreateUpdateFile` which does the following:

- the program reads the **first** match results from `RESULTS.txt`
- appends to the results of each team to a text file `NEWFILE` with the following information: team name, result of match (`W/D/L`), goals for (`GF`), goals against (`GA`)
- for e.g. the data "`MadUnited 2 Chelsand 1`" will result in the following two records to be appended to `NEWFILE`,

  ```
  MadUnited,W,2,1
  Chelsand,L,1,2
  ```

Evidence 1: Your `CreateUpdateFile` program code. **[8]**

Task 2

Amend your `CreateUpdateFile` program code from task 1 so that all the match results are read from `RESULTS.txt`, and the `NEWFILE` updated accordingly.

Evidence 2: Your program code for the amended procedure `CreateUpdateFile`. **[6]**

Task 3

Write program code for a function `ComputeTeamStat` which does the following:

- receives a team name as a parameter,
- searches the file `NEWFILE` for all occurrences of that team,
- calculates and outputs the team's league table information, e.g. for team '`MadUnited`', it may output the following:

Team	**P**	**W**	**D**	**L**	**GF**	**GA**	**GD**	**Points**
MadUnited	3	2	1	0	5	2	3	7

Evidence 3: Your `ComputeTeamStat` program code. **[8]**

Evidence 4: A screenshot showing the output for the team **`Chelsand`**. **[4]**

Task 4

Write program code for a procedure `GenerateTable` which does the following:

- reads the data from files `TEAMS.txt` and `NEWFILE`,
- makes use of the function `ComputeTeamStat` from task 3,
- and output the complete league table information.

Evidence 5: Your `GenerateTable` program code. **[8]**

Evidence 6: A screenshot showing the output for the complete League table. **[4]**

Task 5

Amend your `GenerateTable` program code from task 4 to output the complete League table ordered by the team with the highest points first. If the teams have the same points, then the team with better goal difference (`GD`) will be placed higher.

Evidence 7: Your program code for the amended procedure `GenerateTable`. **[2]**

Evidence 8: A screenshot showing the output for the complete league table. **[2]**