CSC108H Winter 2023 Worksheet: Structured Files

The NHL-data.txt file format:

The file `NHL-data.txt` is a text file that contains the characters:

```
Toronto Maple Leafs\n2\n2\n1\n0\n0\n2\nGrande Prairie Storm\nMontreal Canadiens\n1\n2\n1\n0\n2
```

Interpreting the newline character `\n` as an instruction to move to a new line allows us to visualize the file as:

```
Toronto Maple Leafs
2
2
1
0
0
2
Grande Prairie Storm
Montreal Canadiens
1
2
1
0
2
```

The file contains hockey team names, with each team name followed by the number of points earned in each game in the NHL season so far. Note that the `Toronto Maple Leafs` have played 6 games, the `Montreal Canadiens` have played 5 games and the `Grande Prairie Storm` have not played in the NHL yet. This file is more complicated than the `dictionary.txt` file since the lines can contain either a team name or a number of points, and a line with a number of points belongs to the most recently read team name.

Before writing code that reads a file that does not have a simple structure, it can be helpful to write an abstract description of the file contents. The `NHL-data.txt` looks like:

```
Team Name 1
points from Team Name 1's game 1
points from Team Name 1's game 2
...
points from Team Name 1's game N1
Team Name 2
points from Team Name 2's game 1
...
points from Team Name 2's game N2
Team Name 3
points from Team Name 3's game 1
...
points from Team Name 3's game N3
```

Note that in our example file `N1` is 6, `N2` is 0 (no NHL games yet) and `N3` is 5.

More generally, a `NHL-data.txt` file contains a sequence of team names, with each team name followed by the number of points earned by the team in each NHL game. The number of team names could be 0, 1 or some larger number, and the number of games played by each team could be 0, 1 or some larger number. **We will assume that the team names and number of points earned each have a valid value.**

CSC108H Winter 2023 Worksheet: Structured Files

Writing code to read more complicated files:

The following function is passed a reference to an already opened file that has the same format as the `NHL-data.txt` file. Fill in the boxes below with appropriate `python` code so the function works as described. Before writing code, answer the questions:

1. What value will be read when the end of the file has been reached?
2. How can you determine whether or not a line contains a number of game points?
3. When might it not be possible to determine an average number of points per game?

```
from typing import TextIO

def points_per_game(game_data: TextIO) -> list[list]:
    """Return a list containing the team name and the average number of points
    earned per game for each team in the open file game_data. If the team has
    no games, use None instead of an average number of points.

    >>> input_file = open('NHL-data.txt')
    >>> points_per_game(input_file)
    [['Toronto Maple Leafs', 1.17], ['Grande Prairie Storm', None], \
['Montreal Canadiens', 1.2]]
    >>> input_file.close()
    """
    result = []

    # read a line from the file to set up for the outer while loop condition
    line = game_data.readline().strip()
    while line != '':
        # due to the structure of the file, line contains a team name.

        team_name = [          ]
        # set up accumulator variables for the team.
        games_played = [          ]
        total_points = [          ]

        # read and process game points until: new team name is read or end of file is reached
        [                                                                    ]
        while [                                                              ]:
            total_points = total_points + [          ]
            games_played = games_played + 1
            [                                                                ]

        # add results for team_name to result list
        if [                                                                 ]:
            result.append([team_name, round(total_points / games_played, 2)])
        else:
            result.append([team_name, None])
    return result
```