CSC 201
Lab 6

Files uploaded for scoring that execute with a Python error message will earn **ZERO** points for any function that doesn't execute.

Part 1: Getting Started

Download the directions and Lab6 zipped folder. Unzip Lab 6 folder. You should have 3 files: basketball_stats.py, data1.txt, and data2.txt.

The program is decomposed into functions. Write each function, then test it instead of trying to write the entire program before testing. Add your code below the function's docstring.

When completed, your program will display basketball stats with the data read from a file.

- Execute the program entering data1.txt for the filename. The data from the file is displayed in a nicely aligned table.

```
Enter filename: data1.txt

        Player 0  Player 1  Player 2  Player 3  Player 4  Player 5  Player 6  Player 7
Game 0      14         7         5         2        16        21         4         0
Game 1      10        13         9         6        21        25         5         0
Game 2       8         6         4         3        27        23         2         4
Game 3      11         0        13        12        18         9         4         2
Game 4      16        10        12         7        20        16         3         7
Game 5       9         8         6        10        19        24         3         3
```

In function main, **`points`** is a 2D list of integers representing the points scored by players on a basketball team. Each row stores the points for one game. Each column stores the points for one player.

Part 2: get_team_average function

- Define function get_team_average which accepts one parameter, the 2D list of points. The function should compute and return the average points scored by the team per game (the total points scored divided by the number of games).

- Add code in function main to call get_team_average and print the average with 2 decimal digits and this text:

  `Team's average points per game is XX.XX`

- Save and execute. Check your output for data1.txt and data2.txt. The averages should be 79.50 and 47.29, respectively.

Part 3: get_game_total function

- Define function get_game_total which accepts two parameters: the 2D list of points and a game number (int). The function should compute and return the total points scored one game, the game accepted as a parameter.

- The function print_game_points has already been written to call your function get_game_total. Read the docstring for function print_game_points. Add a call to print_game_points in function main.

- Same and execute. For data1.txt, the call to print_game_points should produce the following table where the points totals are computed by calling function get_game_total.

```
Game   Points
  0      69
  1      89
  2      77
  3      69
  4      91
  5      82
```

Part 4: max_player_points function

- Define function max_player_points which accepts two parameters: the 2D list of points and a player number (int). The function should compute and return the maximum number of points scored by the player, accepted as a parameter, in any one game.

- Function get_int_num from Program Practice 3 is at the top of this file. Read its docstring to recall what the parameters represent. In function main, call function get_int_num to prompt the user to enter a valid player number. The prompt should be 'Enter player number: '.

- Following the call to get_int_num in function main, call function max_player_points and print the player's maximum points. Save and execute using data1.txt. Here's the output to match:

```
Enter player number: 8
Invalid, Try again.
Enter player number: -1
Invalid, Try again.
Enter player number: 7
Player 7's max points in one game is 7
```

Part 5: Test both files thoroughly!

- Execute the program with both data files. The output is on the next page. **Match my output!**

- Thoroughly test the player number entered by the user using invalid integers and each valid integer. (We will assume that the user enters an integer instead of letters.)

Part 6: Upload your work!

- Upload **basketball_stats.py** to Moodle. Upload it as a .py file not pdfs or something else. The code you upload must execute without Python error messages.

Output for data1.txt

```
Enter filename: data1.txt

        Player 0  Player 1  Player 2  Player 3  Player 4  Player 5  Player 6  Player 7
Game 0      14         7         5         2        16        21         4         0
Game 1      10        13         9         6        21        25         5         0
Game 2       8         6         4         3        27        23         2         4
Game 3      11         0        13        12        18         9         4         2
Game 4      16        10        12         7        20        16         3         7
Game 5       9         8         6        10        19        24         3         3


Team's average points per game is 79.50

Game   Points
  0      69
  1      89
  2      77
  3      69
  4      91
  5      82

Enter player number: -1
Invalid, Try again.
Enter player number: 9
Invalid, Try again.
Enter player number: 2
Player 2's max points in one game is 13
```

Output for data2.txt

```
Enter filename: data2.txt

        Player 0  Player 1  Player 2  Player 3  Player 4
Game 0       5        12         3        16        10
Game 1       3         9         2        21        12
Game 2       4        16         1        18         5
Game 3       0        14         5        12        18
Game 4       1        11         3        17        11
Game 5       2        15         2        10        17
Game 6       3        12         6        21        14


Team's average points per game is 47.29

Game   Points
  0      46
  1      47
  2      44
  3      49
  4      43
  5      46
  6      56

Enter player number: 0
Player 0's max points in one game is 5
```