Overview

This homework is due Friday, April 3, 2015 at 11:59:59 PM. You will find a number of files in the zip folder (from the course website) associated with this homework. The files are explained in the relevant sections below.

Points to remember:

- Name your submissions `hw04_part1.py`, `hw04_part2.py`, and `hw04_part3.py`, all lowercase. Otherwise, we will not be able to locate and execute your submissions properly.
- As in Homework#3, expect different input cases and input files in the submission server. Your program must work for all possible inputs, not just the ones you test for!
- Remember to print any input you read immediately as we have done in previous homeworks.
- We will grade you on correct program execution as well as good program structure. We will also continue to check for “excessive collaboration,” so be mindful of this.

Part 1: World cup strikes again

Your solution for this part must be called `hw04_part1.py`. Only submit this file for this part.

You will use the input module called `hw4_util.py` to read a file `all_games.txt` containing all matches in the World Cup. Read this file into a list of lists in which each item in the list is a different game.

```python
import hw4_util
results = hw4_util.read_games( 'all_games.txt' )
print results[-1]
```

For example, given `['Brazil', 3, 'Croatia', 1, 'Brazil']`, we know that the score of the Brazil-Croatia game was 3-1, and Brazil (the last element on the list) won.

Similarly, `['Netherlands', 0, 'Costa Rica', 0, 'Netherlands']` means that the score of the Netherlands-Costa Rica was 0-0, but Netherlands won (presumably by penalty kicks).

Finally, `['Ecuador', 0, 'France', 0, 'Draw']` means that this game was a draw. No one won.

Write a program to do the following:

- Read the name of a country from the user (upper/lowercase should not matter).
- Find and print all the games this country played. Print the total number of games played, wins, losses, draws, goals for (GF), and goals against (GA).
• Print whether the country played in the round of 16, the quarter finals, semi finals, or the final.

The game number will determine the last bullet item above. The first 48 games are group matches, games 49-56 are in the round of 16, games 57-60 are quarter finals, games 61-62 are semi-finals, the 63rd game is the third place playoff, and the 64th game is the final.

Print the highest achievement for the team (only the group they played in).

• Each game is printed as team 1 name (use \texttt{rjust(24)}), team 1 score (\texttt{rjust(4)}), team 2 score (\texttt{ljust(4)}), and team 2 name (\texttt{ljust(24)}). All team stats are printed with \texttt{ljust(6)}.

• Print the final standing of this team if it is one of first, second, third, or fourth place winners.

Below are possible runs of your program:

Please enter a country ==> Algeria
Algeria
All games:

<table>
<thead>
<tr>
<th>Team</th>
<th>Score 1</th>
<th>Score 2</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belgium</td>
<td>2-1</td>
<td>Algeria</td>
<td>Winner: Belgium</td>
</tr>
<tr>
<td>Korea Republic</td>
<td>2-4</td>
<td>Algeria</td>
<td>Winner: Algeria</td>
</tr>
<tr>
<td>Algeria</td>
<td>1-1</td>
<td>Russia</td>
<td>Winner: Draw</td>
</tr>
<tr>
<td>Germany</td>
<td>2-1</td>
<td>Algeria</td>
<td>Winner: Germany</td>
</tr>
</tbody>
</table>

Scores:

Games Win Lose Draw GF GA
4 1 2 1 7 7

The highest achievement in 2014 World cup was round of 16

Please enter a country ==> ENGland
ENGland
All games:

<table>
<thead>
<tr>
<th>Team</th>
<th>Score 1</th>
<th>Score 2</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>England</td>
<td>1-2</td>
<td>Italy</td>
<td>Winner: Italy</td>
</tr>
<tr>
<td>Uruguay</td>
<td>2-1</td>
<td>England</td>
<td>Winner: Uruguay</td>
</tr>
<tr>
<td>Costa Rica</td>
<td>0-0</td>
<td>England</td>
<td>Winner: Draw</td>
</tr>
</tbody>
</table>

Scores:

Games Win Lose Draw GF GA
3 0 2 1 2 4

The highest achievement in 2014 World cup was group games
Please enter a country ==> germany
germany
All games:
    Germany 4–0 Portugal  Winner: Germany
    Germany 2–2 Ghana  Winner: Draw
    USA 0–1 Germany  Winner: Germany
    Germany 2–1 Algeria  Winner: Germany
    France 0–1 Germany  Winner: Germany
    Brazil 1–7 Germany  Winner: Germany
    Germany 1–0 Argentina  Winner: Germany

Scores:
Games Win Lose Draw GF GA
7 6 0 1 18 4
The highest achievement in 2014 World cup was finals
This country placed in the first place

Please enter a country ==> Turkey
Turkey
All games:
Scores:
Games Win Lose Draw GF GA
0 0 0 0 0 0
The highest achievement in 2014 World cup was not making it to the tournament

Here are some additional hints to get you started. You will need a loop to iterate through all the games one by one, checking whether the current team is actually the team we are searching for. Once you find it, print it out.

Next, add some variables to count the total number of wins, draws, and losses. Remember the team name can appear as the first or the second team!

When you are done with all of this, the final step is to look at the team standing. Keep track of the highest index in the list in which you find a game for this team. Using this index, you can figure out where this team has played.
Part 2: Functions that change list contents

In this part, you will use two inputs corresponding to two individuals walking in a grid of \((x,y)\) coordinates. You will know their starting point and the path they took. Here is example input:

Wallace,2,2,up,left,stay

which means that:

- Wallace was initially at point \((2,2)\).
- Then, he walked up at time 1 to position \((2,3)\).
- Then, he walked left at time 2 to position \((1,3)\).
- Then, he stayed at his current position \((1,3)\) at time 3.

Your job is to read two strings corresponding to the two paths using `raw_input()`, then use this information to print the location of each individual and the grid distance between them (total number of grid segments in between) at each time point. As an example, the distance between \((5,4)\) and \((2,3)\) is given by \((5-2)+(4-3)=4\).

Finally, print the maximum and minimum distance between the two individuals during their walk.

You will want to use the `split()` function to process the two input strings.

Here is sample output:

```
Enter the first path ==> Wallace,2,2,up,left,stay,up,right,stay,up,right
Wallace,2,2,up,left,stay,up,right,stay,up,right
Enter the second path ==> Gromit,5,4,down,down,left,stay,left,left,down,down
Gromit,5,4,down,down,left,stay,left,left,down,down
Wallace   Action   Gromit  Action   Distance
(2,2)    initial   (5,4)  initial   5
(2,3)    up        (5,3)  down     3
(1,3)    left      (5,2)  down     5
(1,3)    stay      (4,2)  left     4
(1,4)    up        (4,2)  stay     5
(2,4)    right     (3,2)  left     3
(2,4)    stay      (2,2)  left     2
(2,5)    up        (2,1)  down     4
(3,5)    right     (2,0)  down     6
The minimum distance between them is 2
The maximum distance between them is 6
```

**Remember:** Correctness and good program structure will be a big part of your grade. You must use a list for an individual’s current location, i.e \([1,2]\). You must use a function to find the next location of an individual by changing the input location and the current command (up, down, left, right, stay). Your function must not return a value, but instead modify the input location list.

Do your best to match the spacing in the output. Your columns must be aligned. The above output uses 11 spaces for each column.

Once you are sure your program works correctly, turn in only your code (`hw04_part2.py`).
Part 3: A bit of repetition

Your solution for this part must be called hw04_part3.py. You cannot use any modules for this part.

Write a program that:

- Reads an input string from the user.
- Reads a maximum line length (which will be an integer).
- Prints the input string on exactly 10 lines up to the given maximum line length, wrapping the remainder of the string from one line to the next line (don’t worry about breaking on spaces).

Below is an example run of the program.

Please enter a line ==> Do I look like a guy with a plan?
Do I look like a guy with a plan?
Please enter a line length ==> 60
60
Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan?

EXTRA CREDIT: Modify your code to pay attention to space characters, breaking on and omitting all spaces at the end of each line.

Below is then an example run of the program.

Please enter a line ==> Do I look like a guy with a plan?
Do I look like a guy with a plan?
Please enter a line length ==> 60
60
Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan? Do I look like a guy with a plan?