COSC 1336 Lab: 9

Relevant reading: Chapter 8

Due: Nov. 20, 2:30 pm

(Late date: Nov. 27, 2:30 pm)

50 Points

Problem 1. [10 points] For this problem, you will complete a program that allows the user to enter a date and tells the user on which day of the week the date fell or will fall. Download the file `zellers.py` (so named because the algorithm for converting a date to a day of the week is called Zeller's congruence). In it you will find a bunch of functions defined for you; the ones you need to worry about are `read_date`, `calc_day_index`, and `date_to_str`.

All you have to do for this problem is fill in the `main` function, and if you use the provided functions correctly, this means writing less than 1 lines of code. Once you have finished reading the description of the program you should write, read the comments above the provided functions to find out how to use them. The challenge of this problem is largely about reading and understanding how to use provided code, so don't start implementing until you understand what you're looking at (and consider asking questions if you are confused).

Your program's interaction should look something like this:

```
Enter the number of the month (zero to quit): 77
Input must be between 0 and 12.  Try again: 7
Enter the day: 7
Enter the year: 2012
07/07/2012 is/was a Saturday

Enter the number of the month (zero to quit): 11
Enter the day: 122
Input must be between 1 and 30.  Try again: 12
Enter the year: 2012
11/12/2012 is/was a Monday

Enter the number of the month (zero to quit): 0
```

Problem 2. [30 points] Download the files `WorldSeriesWinners.txt` and create a new file named `world_series.py`. The text file contains a list of all of the winners of the U.S. baseball world series, starting with the year 1903. In the file `world_series.py`, write the following functions. I recommend that you first read through the whole problem, so you know where things are going, but then implement and test each function thoroughly before continuing to the next part.

a. **[5 points]** Write a function called `read_winning_teams` has no parameters and returns a list of all of the data in the file. Since some teams have won the series multiple times, they should appear multiple time in your list. Test the function well before you continue by comparing the list the function returns to the data in the actual file.

b. **[5 points]** Write a function called `get_wins` which receives the list of winner for each year (that is, the list created by the `read_winning_teams` function), and a team name, and returns a list of the years in which the team won the world series. Remember that the list starts with the year 1903.

c. **[5 points]** Write a function called `copy_no_duplicates` which receives a list and returns a copy of the list, with duplicates removed, and sorted (using the list `sort` function). Make sure that you do not modify the original list.

d. **[5 points]** Write a function called `print_menu` which receives a list of baseball teams with no duplicates, and generates a menu that allows the user to choose a tema by specifying a number. It should produce output that looks like this (but with all the teams listed):

```
Choose a team and specify its number:
0: Anaheim Angels
1: Arizona Diamondbacks
...
26: Toronto Blue Jays
27: Washington Senators
-1 to Quit
```

e. **[5 points]** Write a function called `get_user_choice` that uses the `print_menu` function and an input validation loop to read in and return a valid choice from the user. Valid choices are -1 (to quit) or a number between 0 and one less than the number of teams.

f. **[5 points]** Finally, write the `main` program that pulls all the pieces together. The function should have a sentinel loop, where the user's input is read in using the `get_user_choice` function, and the sentinel value is -1. When the user's choice is the index of a team, the program should calculate and display a list of the years in which the team won the world series. After the result has been displayed, the program should wait for the user to press enter before continuing, or it will be difficult to read the result, since the menu is so long. So the completed program will look something like this (but with the complete menu):

```
Choose a team and specify its number:
0: Anaheim Angels
...
27: Washington Senators
-1 to Quit
Your choice? 25

The St. Louis Cardinals won in the following years: [1925, 1930, 1933, 1941,
1943, 1945, 1963, 1966, 1981, 2004]

Press enter to continue...

Choose a team and specify its number:
0: Anaheim Angels
...
27: Washington Senators
-1 to Quit
Your choice? -1
```

Problem 3. **[10 points]** For this problem, you will use several of the functions from the last problem in conjunction with one more new function to create a bar graph of the world series data. Download the files `draw_functions2.py` and `world_series_graphics.py`, and do the following:

a. Copy and paste the following three functions from your solution to problem 2 to the indicated spot in `world_series_graphics.py`:

- `sorted_copy_no_duplicates`
- `read_winning_teams`
- `get_wins`

b. **[10 points]** Write a function called `get_data` which receives the list of winning teams for each year, along with the list of teams with duplicates removed. The function should return a list of tuples, where each tuple has a team name in the first position, and the number of wins the team has in the second position. That is, the function should return a list the beginning of which looks like this:

```
[('Anaheim Angels', 1), ('Arizona Diamonbacks', 1), ('Atlanta Braves', 1),...
```

c. Once you've completely debugged your `get_data` function, uncomment the three lines in the file that actually create the bar graph. If you've written all the functions correctly, the bar graph should appear. You are welcome to take a look at the `draw_bar_graph` function if you are interested.

For this lab, you should submit the following files:

- `zellers.py`
- `world_series.py`
- `world_series_graphics.py`