COP 3223 Program #4: Hurricane data summary

Due on Webcourses: Tuesday, October 18, 2016, 11:00 PM

Program: Hurricane data summary (hurricane.c)

So far in 2016, we have had five storms that achieved hurricane status: Alex, Earl, Gaston, Hermine, and Matthew. NOAA tracks the status of hurricanes and tropical storms and makes the recorded data is available to the public. You have been asked by the local weather station to write a program that will summarize the tracking data for any selected storm.

The data file for each storm encodes data in the following format. The first line contains a single integer which indicates the number of remaining lines of data in the file. Each line of data contains three integers: month, day, and wind speed in knots. For example, the data file for hurricane Alex is as follows:

```
9
01 13 45
01 14 50
01 14 60
01 14 75
01 14 75
01 15 70
01 15 65
01 15 60
01 15 60
```

This file indicates that there are ten data points for Alex. January 13, 2016 was the first day that Alex achieved named storm status and it had a wind speed of 45 knots. On January 14, 2016, four data points were recorded with wind speeds of 50, 60, 75, and 75 knots, respectively. On January 15, 2016, four data points were recorded with wind speeds of 70, 65, 60, and 60 knots, respectively. The last day on which Alex had named storm status was January 15, 2016. The maximum wind speed achieved by Alex was 75 knots. The average wind speed of Alex, based on all recorded data points is 62.22 knots.

Your program should read in the data for a given storm and print out a summary of that storm including the start date, end date, average speed in knots, maximum speed in knots, and maximum category achieved.

Your solution must include a main program and four functions: `read_data()`, `print_summary()`, `get_category()`, and `print_month()`. Details of the expected structure of your program are as follows:

Program: `main()`

Your main program should do the following three things:
1. Open a link to an input file called `input.txt` which contains the tracking data for a single storm. ***Important note: Please ensure that your program reads an input file titled*** `input.txt`***. If your program is set up to read a file by any other name, it will be unable to find the input file during grading and you will lose 40% or more of your grade (due to incorrect output) even if the rest of your program is logically correct.***
2. Call the function `read_data()` that reads the data from `input.txt` and prints out a summary of the storm data.
3. Close the link to the input file.

Function: `read_data()`
Pre-conditions: The input to this function is a file pointer linked to the input file to be read.
Post-conditions: This function does not return anything.
Actions:
Function `read_data()` should do the following:
1. Read data from the input file, saving the maximum wind speed and the start and final month and day.
2. Calculate the average wind speed over all data points.
3. Call function `get_category()` to get the maximum category achieved by the storm.
4. Call function `print_summary()` to print out the summary data.

Function: `get_category()`
Pre-conditions: The input to this function is a single integer value representing a wind speed in knots.
Post-conditions: This function returns an integer value denoting the hurricane category represented by the given wind speed.
Actions:
Function `get_category()` should do the following:
1. Given a wind speed in knots, return the appropriate hurricane category according to the Saffir Simpson Hurricane Wind Scale.
See http://weather.unisys.com/hurricane/atlantic/2016/ for the Saffir Simpson Hurricane Wind Scale. Keep in mind that the data stores wind speeds in knots.

Function: `print_month()`
Pre-conditions: The input to this function is a single integer value between one and twelve representing a month.
Post-conditions: This function does not return anything.
Actions:
Function `print_month()` should do the following:
1. Given an integer value representing a month, print the name of the month. Does not print an end-of-line symbol after the name.

Function: `print_summary()`

Pre-conditions: This function receives six integer values (start month, start day, final month, final day, maximum speed in knots, and maximum category) and one double value (average speed in knots over all data points) as input.

Post-conditions: This function does not return anything.

Actions:

Function `print_summary()` should do the following:

1. Print a summary of the storm data as shown in the example below. Use the function `print_month()` to print the correct name of the month for the given the integer value.

Sample Program Run #1: output for data file `alex.txt`
```
Start day: January 13, 2016
Final day: January 15, 2016
Average wind speed:  62.22 knots
Maximum wind speed:  75 knots
Maximum category: 1
```

Sample Program Run #2: output for data file `gaston.txt`
```
Start day: August 23, 2016
Final day: September 4, 2016
Average wind speed:  65.42 knots
Maximum wind speed: 105 knots
Maximum category: 3
```


Deliverables

Please submit a file titled hurricane.c containing your solution to this problem to Webcourses by Tuesday, October 18, 2016 at 11:00 PM. Please submit your file to both the "Homework 4" assignment and the "Homework 4 Peer Review" assignment.

Some notes:

1. Please ensure that your file includes header comments which include a good description of your entire program.

2. Each function should be preceded by comments that give the pre-conditions, post-conditions, and actions of that function.

3. All of your code should be commented. A good measure of the effectiveness of your comments is whether I can understand the structure of each main program and function by the comments alone (e.g. can I tell what your program and functions do if I deleted all of the code and only had your comments).

4. All variables declared within a function must be declared at the start of your function. All variables in the main program must be declared at the start of your main program.

Program: **hurricane.c**