Assignment #5

Introduction to C Programming – COP 3223

Objectives

1. To reinforce the use of loops.
2. To learn how to read information from files.

Introduction: Pirate Time

Your friend has not stopped talking about how cool pirates and how awesome it would be if they could be a real-life pirate captain! To amuse your friend, you have decided to create a series of programs about pirates.

Problem: When to Set Sail? (settingsail.c)

Your friend is almost ready to leave Spain to search for treasure in the Caribbean! To make sure that the maximum number of crew members make it through the trip safely, your friend will carefully select which months to travel the ocean during.

In this program, you will examine a file of weather data. Ask the user for the name of this file. Then, ask the user for the temperature range that their crew sails best in. Calculate the percentage of days in each month that fall into that range. Pirates are not the most skilled at recording date, so a file of weather data may not contain information for every day in every month. Instead, calculate the percentage by dividing the number of days in range by the number of days that there is data for.

The month with the highest percentage is the month that your friend should pick to begin their journey for treasure!

Arrays are not expected to be a part of this program, except for the name of the input file. Do not attempt to store all the values in the file in your program.

Input Specification

1. The lower bound temperature will be an integer between 0 and 80
2. The upper bound temperature will be an integer higher than the lower bound and less than 120
3. The file name will be a string less than 30 characters in length

Input File Specification

There will be 12 months of weather data available. Since months may not have complete information, each month will begin with a header number (*n*) to indicate how many temperatures are available for that month. The month will then have *n* real numbers representing the temperature for each day.

Output Specification

Output to the screen. Show the user the percentage of days in range for each month. Then, print the best month for your friend to begin their voyage.

```
Month X: YY.Y percent of days in range.
You should leave for the Caribbean in month Z!
```

Output Sample

Below is a sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

The corresponding sample input file is available on the webcourse.

```
Tell me about your crew's preferred temperature for sailing:
What is the coldest temperature they can sail in?
60
What is the hottest temperature they can sail in?
80
Please enter the name of the weather data file:
temperatures.txt

Month 1: 93.3 percent of days in range.
Month 2: 20.0 percent of days in range.
Month 3: 58.1 percent of days in range.
Month 4: 100.0 percent of days in range.
Month 5: 100.0 percent of days in range.
Month 6: 50.0 percent of days in range.
Month 7: 64.5 percent of days in range.
Month 8: 9.7 percent of days in range.
Month 9: 60.0 percent of days in range.
Month 10: 100.0 percent of days in range.
Month 11: 93.3 percent of days in range.
Month 12: 83.9 percent of days in range.

You should leave for the Caribbean in month 4!
```

Deliverables

One source files – *settingsail.c* – is to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Code::Blocks. Your program should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure you include comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.