

## **Introduction to C – Programming #3** *(Is the Air Running?)*

**Due Date: 03/21**

### **Objective**

To give students practice at writing a program reading input from a file that uses an array.

### **The Problem**

A client is interested in exactly what percentage of the time her air conditioning is running. One way to measure this is to keep a temperature logging device right by an AC vent. The device measures the temperature every 30 seconds. If the temperature drops at least .5 degrees Fahrenheit between readings, it's a good bet that the AC unit just turned on. Likewise, if the temperature rises at least .5 degrees Fahrenheit between readings it's likely that the AC unit just turned off. (For example, if the reading at 1:30:00pm was 74.7 degrees F and the reading at 1:30:30pm was 74.1 degrees F, then we would assume for the 30 seconds starting at 1:30:00pm and ending at 1:30:30pm, the AC was on. If there is no big rise or fall in a 30 second interval, we assume the state of the AC was whatever it was in the previous 30 second interval. *We will assume that the AC is always off at midnight.*

You will be given the temperature readings from one logging device for a whole day (24 hours). Using this information, you must calculate the percentage of time the air was on for the day and produce a bar graph with hourly data.

### **Input File Format (temp.txt)**

This file will have exactly 2881 doubles, one per line, indicating readings of temperatures (in Fahrenheit) starting at midnight and ending at midnight the next day. The readings are taken every 30 seconds. The start of a file might look like this:

```
78.6  
78.6  
78.5  
78.5
```

You have to create your own file **temp.txt** to test your program.

### **The Output**

First, output a line with the following format:

The AC was running X percent of the time.

where X is a percentage rounded to 2 decimal places, corresponding to how often the AC was running throughout the whole day.

After this line of output, skip a blank line and output a bar graph with a similar format to the one in the posted file.

Each bar corresponds to the percent of time the AC was running during that hour. Since the bars are shown with 5% increments, only display a star if the percentage for that hour equals or exceeds the marked percentage. Here's the basic idea in how to create it.

First, store the data in an array of size 24, where each entry indicates the percentage of time the AC was running in that hour. (For example, the entry in index 0 stands for the amount of time the AC was running from midnight to 1 am.)

Now, print each row in order, first the row for 100%, then 95%, then 90%, etc. all the way down to 5%. While you are printing a row, for each column, you must ask yourself the question, "Should I print a star at this location, or a space?" A star gets printed if for that hour, the percentage was equal to or greater than that of the row that is currently getting printed. A space gets printed otherwise. Consider the following example:

If the array stored 66% in index 0, 40% in index 1 and 53% in index 2, then imagine printing the row that corresponds to 50%:

You would print a STAR for the first character, since  $66 \geq 50$ .

You would print a SPACE for the second character, since  $40 < 50$ .

You would print a STAR for the third character, since  $53 \geq 50$ .

### **Output Sample**

Posted as separate attachments.

### **Deliverables**

You must submit your solution to the problem, *aircond.c*, over WebCourses.

### **Restrictions**

Although you may use other compilers, your program must compile and run using Codeblocks or DevC++. 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 ample comments throughout your code describing the major steps in solving the problem.

### **Grading Details**

Your program 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 to Codeblocks or DevC++. (If your program does not compile in this environment, you will get a **sizable** deduction from your grade, likely to be over 50%)