

Homework Assignment #4

Course: COP 3223C – Intro to Programming with C

Semester: Fall 2016

Credit Value: 5% of Final Grade

Due Date: October 26, 2016

Loops and Arrays

Description: You are building a mini-weather station in your apartment in Florida and want to store a series of ambient temperature readings every hour for 24 hours, and then find the average, maximum and minimum values for that day period. We start measuring at midnight (time 0:00) until midnight the following day (time: 24:00, or 0:00 the next day). Actually, that makes it 25 readings, right?. We want to store the data collected in an array of integers called `temp[]`, where index 0 (`temp[0]`) contains the reading at midnight, index 1 (`temp[1]`) contains the reading at 1:00 AM, index 2 is 2:00 AM and so on.

However, while we write and test the program we won't have access to the actual temperature sensor or the data acquisition system. So, we have to simulate the data collection. We can do this by generating random numbers between 60 and 100 (for degrees Fahrenheit) and inserting them into the array sequentially until we fill up the array. (Nevermind that the numbers won't be gradually increasing, peaking in the afternoon and then decreasing, as one would expect ambient temperature to do.) Then, we calculate the average, maximum and minimum values and print them out as a simple table.

So, you will need to design a loop within `main()` to populate the array, one cell at a time. Moreover, you will need to write **two** programmer-defined functions:

- 1) The first function called `get_value()` will return a random integer between 60 and 100. This function will be called from `main()` and the value returned will be inserted into the proper location in the array.
- 2) The second function called `calc_results(int [], int)` will accept the entire array once it has been completely populated, and will identify the maximum and minimum temperatures, as well as calculate the average temperature for the day. Note that you will have to convert integer values into floating point values to calculate the average. It will also print out the 25 temperatures in a table as well as the three computed values.

The output should look like this: (assume October 31, 2016 is the day for which the values were collected):

Temperature Conditions on October 31, 2016:

Time of Day	Temperature in degrees F
0	85
1	80
2	97
3	90
4	68
5	75
6	77
7	98
8	97
9	62
.	.
.	.
.	.
24	89

Maximum Temperature for the day: <whatever> Degrees F

Minimum Temperature for the day: <whatever> Degrees F

Average Temperature for the day: <whatever.whatever> Degrees F