Homework Assignment #5
Course: COP 3223C – Intro to Programming with C
Semester: Fall 2016
Credit Value: 6% of Final Grade
Due Date: November 14, 2016

Sequence of Events Recorder

Description: Unexpected destructive failures of expensive equipment can be very frustrating to designers and maintenance personnel because they don't often have enough information to allow them to determine what caused the failure. Because failures are almost always unexpected, and typically result in complete loss of the device, doing a port-mortem analysis is often difficult, leaving them to do no more than a visual inspection of the "corpse". Examples of these are the data black boxes in all commercial aircraft that record the last several minutes of the critical variables of the aircraft's operation just before the accident. Therefore, in failure analysis, it is quite useful to know what the values of critical parameters were just prior to the catastrophic failure. This is the purpose of a sequence-of-events recorder.

Sequence-of-events recorders are designed to capture the last few seconds, minutes, or hours of the operation of the device of interest. They are generally located externally to the device to avoid them being destroyed along with the device. Of course, this is not possible for aircraft black boxes, so they make them almost indestructible instead. By recording only the last few seconds/minutes/hours of operation, the memory requirements can be managed rather well, especially when many variables are to be logged and/or they are to be sampled at a very high rate of frequency. These recorders, therefore, will overwrite old data with new data, always having in memory the critical last several seconds, minutes, or hours of the device's operation before the failure.

You work for a company that manufactures anti-lock braking systems for disk brakes for the major automobile manufacturers. Recently, there have been several incidents in which the anti-locking mechanism has failed, resulting in accidents on wet roads. While no one has yet been injured, your company takes these incidents seriously and wants to resolve any issues before anyone does get hurt. Nobody knows exactly how these devices have mis-operated, but several of the drivers involved in such accidents have reported that the normal clack-clack-clacking of the ABS brakes was replaced by a bing-bing-bing, at a much slower frequency. The engineers are baffled as to what might have caused that, but they do know it isn't normal.

So, you have been assigned to design a sequence-of-events recorder to capture the last 10 seconds of operation of the car. There are several sequence-of-events recorders on the market, but none are specially suited to your high frequency application. So, you have to write the code yourself. Your program is to acquire the values of several variables during one time (called a *slice of data*) and save them to circular array of structures that saves the last 10 seconds of data acquired. The sampling frequency has been specified by the design engineers to be 10 Hertz (10 data slices per second), as auto accidents only take a few seconds to happen.

The system should repeatedly acquire the values of the following variables during the operation:

1) **Brake fluid pressure** to the brake unit: between 20 and 30 PSI.
2) **Electrical power** to the braking unit: between 10 and 100 watts.
3) **Driver brake pressure** (applied by the driver): between 30 and 40 PSI.
4) **Pressure** applied **to the disk**: 10 to 15 PSI.
5) **Time in seconds**.

Specifically, your assignment is to design and write a program that captures the time-tagged data slices of the variables above and saves them to a circular array of structures. Use a sampling frequency of 10 Hertz. Circular arrays overwrite the oldest slice of data with the newest, thereby keeping the last several slices leading up to the accident, which is what you want to do. When an accident occurs (or the car is turned off), the program writes the last several slices of data to an external file. Because you'll only be testing the system for the purposes of this assignment, you can create the values using a random number generator. Save these values to an array of structures that contain a member for each one of the stored variables, plus one more for the time. The time can be assumed to be sequential, starting from 0 and going until the car is turned off normally through the operation of the key … or it crashes. The array is to contain the last 10 seconds' worth of data (100 data slices). Then, write a second program (separate from the one above) that reads the external file and prints it to screen. The output to screen should look something like this:

Time (secs)	Brake fluid pressure	Pressure on disks	Driver brake pressure	Electrical Power to unit
.1	25	15	32	70
.2	20	10	33	72
.3	21	11	32	71
.4	23	10	30	75
.5	24	12	33	78
.6	21	10	36	70
.7	23	12	34	75
.8	22	11	38	78
.9	25	13	35	73
1.0	24	15	32	72
1.1	23	12	31	80
1.2	25	16	33	73
1.3	27	17	36	69

Where it prints the entire 10 second run of the values of each of the variables, at each 0.1 second.

Notes

1. You **must** use the following elements in the program:
    a. A `struct` structure
    b. An array of size 100.
    c. A random number generator (`rand()`)
    d. At least two user-defined functions
    e. The `typedef` data specifier for the structure.
2. Put the data acquisition process in a `for` loop with exactly 1177 cycles. Assume that when it ends, the car crashed or the driver got to his/her destination and turned the engine off. Use that as the signal to write the collected data to the file.
3. Use integers for the values of the variables, even though floating point values would be much more realistic in this application. It’s much easier than generating floating point numbers.
4. Record the time in intervals of 0.1 seconds.
5. Comment your file adequately. Indent and space adequately.