COP2220 Spring 2022 Project Phase I 4-16-22

**This is a reduced set of specs form the original version of this project. The reason for the reduction is that out term will end in about 14 days. Do your best. Your effort will be taking into consideration.**

**This project is due May 1th, 11:59 pm!**

**You can submit your program (dead or alive) by the 3rd  11:59pm, with a penalty of 10 points per day.**

**DO NOT submit earlier if not complete…will be rejected and not graded. You must attempt all specs. Email for help whenever you encounter difficulties. I will clarify the specs for you and will always refer you to a session that covers what you need. Do not miss any video from now 4-16-22 on and do not ignore the latest sessions.**

**Important! You must work on this project alone. Your grade will be pending an online meeting with me when you will be quizzed on how you coded it.**

**Required: Part I**

The specs side:

You are tasked with the development of an irrigation controller system simulator.

This simulator should be capable of:

-prompting the user to enter date and time of the day (will be done manually first and then automatically via C built-in function from the time library).

-prompt the user to enter the number of zones that need to be managed.

-set a starting date and time for the irrigation to start.

-select how many minutes each zone should get.

-run the different zones concurrently and report on the start and end of each zone operation.

-implement a rain censor. Sensor limits are 0 to 2” of rain. This limit should be set by the user and if reached, the irrigation should not start or stop if already started.

-user should be allowed to reconfigure all the parameters if needed. This will require a configuration menu in the controller.

The coding side:

-your irrigation controller will be coded as a structure with the following attributes (more may be needed as your design progresses):

---current date

---current time.

---rain sensor value.

---number of zones

---sub structure zone: this substructure has the following attributes

---operation date (current or future)

---start time

---end time

---total minutes in operation

---sequence (i.e. if 7 zones then a number between 1 to 7)

---skipped (Boolean) a zone can be skipped after a schedule is set.

--state: on or off (set to on when start time is reached and until the end time)

What to do?

-declare your structure Controller and within it your structure Zone.

-call a function that displays the current date and time and a menu based on the following:

-1-Set date and time

-2-select number of zones

-3-configure operation of all zones (start date, start time, end time)

-4-set the rain sensor limits increment of .5 inches to a max of 2 inches)

-5-schedule the operation.

-6-reset system

-if the user select option 1, then call a function that clear the screen and display the date and time from the system and display them. This function is called for every screen that is displayed to the user.

-if user select option 2, then call a function that prompts the user to enter the number of zones, then using malloc, declare an array of size that number using malloc. This function should make a pointer to this structure available back in main for access. All attributes for this structure ‘zone’ must be initialized.

-if user selects option 3, call a function that prompts the user to enter or allow values to be generated (i.e. time) for all attributes and for each zone.

-if user selects option 4, call a function where the rain sensor should be activated with a random number generator and the activation should be on within the period scheduled for any and all zones (note that not only you have to generate a value for the rain sensor but you have to generate a random activation point and length…in this case, the value of the rain sensor may affect one or more zones…to make it simpler, the operation of all zones must stop if the value set by the user in the sensor is set 0-2” of course a value of zero means no operation).)

-if user selects option 5, then call a function that display the operation of all zones. Note that only one zone of all selected zones must be set to active. You screen must be refreshed to simulates showing the state of all zones with all their attributes. The display should always display the current date and time and the state of the rain sensor.

-if user select option 6, then the system revert to a default state where current date and time need to be reset and all options (1-5 can be repeated).

**Extra credits: Part II**

-The default state of the whole system must be loaded from a file.

-The user configuration data must be saved to a file and loaded every time the program is restarted.

-If user selects option 6, then the user configuration file must be deleted and the default file loaded.

-Use lists only.

**Reminder! All specs are subject to update as we progress toward the end.**