Flood Sensor Alert Simulation

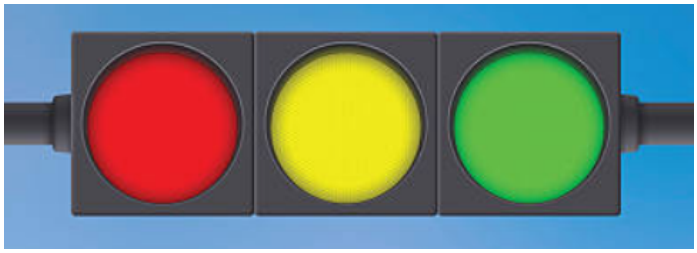
Exercise in Functions and Parameter Passing

The Problem.

The city of Manila is no stranger to flooding. With its low-lying terrain and dense population, the city is prone to flash flooding, especially during the rainy season.

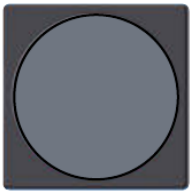
The city has launched a comprehensive flood management program to mitigate the risks and ensure public safety. One key component of this program is installing a network of flood sensors along Taft Avenue, a major thoroughfare that runs through the city's heart. These sensors detect the water level and transmit the data to a central system, using the traffic lights to indicate the flood depth.

Traffic lights Visual Alert



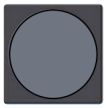
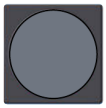
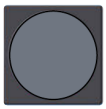
The system uses a combination of red, yellow, and red lights to indicate the flood depth. As mentioned, a corresponding set of light combinations is based on the sensor depth. We shall represent a light switched on with a value of one(1) and zero(0) for off.

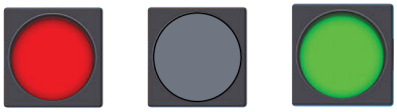
Example:



The red light is set to **1 (ON)**. The Red light is set to **0 (OFF)**

Sensor Data and Light Combination

Sensor Reading	Red	Yellow	Green	Flood Depth
0	 OFF	 OFF	 OFF	Zero (No Flood)
1	 OFF	 OFF	 ON	1 foot
2	 OFF	 ON	 OFF	2 feet
3	 OFF	 ON	 ON	3 feet

4	 ON OFF OFF	4 feet
5	 ON OFF ON	5 feet
6	 ON ON OFF	6 feet
7	 ON ON ON	7 feet and above

Goal

Your task is to implement three (7) functions.

1. getSensorValues() - prompt the user for sensor values
2. displayHeader() - Print the header portion of the output
3. displayDetails() - Print the detail lines of the output
4. isGreenLightOn() - determine if the Light is ON or OFF for a given sensor value

5. isYellowLightOn() - determine if the Light is ON or OFF for a given sensor value
6. isRedLightOn() - determine if the Light is ON or OFF for a given sensor value
7. displayFooter() - Print the header portion of the output

Given

You are provided with the following files.

- **floodSensor.h**
    - Should NOT be modified
    - Contains function prototypes and constant definitions

- **floodSensorMain.c**
    - Should NOT be modified
    - Contains code to produce sample output.

- **GROUPNAME-floodSensor.c**
    - Contains a skeletal code that you need to complete.
    - Contains comments that serve as instructions and documentation
    - Modify the filename and replace LASTNAME with your last name.
    - This file will be submitted through animospace.
    -

To debug and test your program using a user-defined module (.h) and with the main() in a separate file, you need to do the following:
1. You must have all three files above in the same folder (directory).
2. Then, use the following instructions to compile: **gcc -Wall -std=c99 floodSensorMain.c** This will produce a.exe. Note that there is no need to compile GROUPNAME-floodSensor.c separately.
3. Run in the command prompt as usual.
For example, given a.exe generated from step 2, type **a** in the command prompt.

Sample Output

```
❯ ./a.out
Enter sensor reading for Vito Cruz: 1
Enter sensor reading for Quirino Ave: 4
Enter sensor reading for U.N. Ave: 6
Enter sensor reading for Pedro Gil: 3


   Location     Sensor Visual
  along Taft    Value  Alert
-------------- ------- ------
     Vito Cruz    1     001
   Quirino Ave    4     100
      U.N. Ave    6     110
     Pedro Gil    3     011
                ------
         Total   9.00
       Average   2.25
```

```
❯ ./a.out
Enter sensor reading for Vito Cruz: 2
Enter sensor reading for Quirino Ave: 5
Enter sensor reading for U.N. Ave: 0
Enter sensor reading for Pedro Gil: 7


   Location     Sensor Visual
  along Taft    Value  Alert
-------------- ------- ------
     Vito Cruz    2     010
   Quirino Ave    5     101
      U.N. Ave    0     000
     Pedro Gil    7     111
                ------
         Total  16.00
       Average   4.00
```

Technical Constraints

The following restrictions apply when formulating the algorithm and C program implementation.

- The algorithm should only outline the general steps to simulate student crossing.
- Limit the implementation to the concepts discussed in class. These include concepts, Variables, Operators, Expressions, the Assignment Statement, Functions and Parameter Passing.
- Do not use **conditional statements, loops, and string arrays**

Other Requirements

- To be submitted are the following:
    - Source codes - **GROUPNAME-floodSensor.c**
    - Screenshot of the compilation and execution of the application
    - **DO NOT submit** the files floodSensor.h and floodSensorMain.c