Alan Bernal

Pseudocode:

Output the name and data of the assignment.

Extract value from the .obj file

Process the values into dataArray, vsum the values, find the max and min.

Used the bubble sort algorithm to organize dataArray in order.

Use the array values to find the median according to the length of the array.

Output the values with the fprint function.

Code:

|  |
| --- |
| /\*  \* HW Assignment: CPGM0  \* EEL-4746 Fall 2025  \* Alan Bernal  \* Date: 10/8/2025  \* This is my first EEL-4746 C Program (fixed version)  \*/  // Standard Includes  **#include** "driverlib.h"  **#include** <stdint.h>  **#include** <stdio.h>  //Include file for BCUART function  **#include** "HAL\_UART\_4746.h"  // Function Prototypes  **void** **GPIO\_init**();  //Function object file  **extern** uint8\_t **myData**(uint8\_t cmd, uint8\_t \*dataValue);  // Main Function  **void** **main**(**void**){  //Define Local Variable  **char** buffer[100];  uint8\_t dataValue;  uint16\_t i, j;  uint16\_t vCount = 0;  uint32\_t vSum = 0;  uint8\_t vMax = 0;  uint8\_t vAvg = 0;  uint8\_t vMedian = 0;  uint8\_t vMin = 255;  uint8\_t dataArray[256];  //WDT  **WDT\_A\_hold**(WDT\_A\_BASE);  //Initialize LED0 and set it low  **GPIO\_setAsOutputPin**(GPIO\_PORT\_P1, GPIO\_PIN0);  **GPIO\_setOutputLowOnPin**(GPIO\_PORT\_P1, GPIO\_PIN0);  //Initialize and Configure UART  **UART\_initGPIO**();  **UART\_init**();  //Activate New Port Configurations  **PMM\_unlockLPM5**();  **sprintf**(buffer, "Alan Bernal \r\n");  **UART\_transmitString**(buffer);  **sprintf**(buffer, "My Seed Number is z\r\n");  **UART\_transmitString**(buffer);  **sprintf**(buffer, "FALL 2025 \r\n");  **UART\_transmitString**(buffer);  **sprintf**(buffer, "Problem 2-A \r\n");  **UART\_transmitString**(buffer);  **sprintf**(buffer, "The results are: \r\n");  **UART\_transmitString**(buffer);  //--------------CODE------------------//  //Get Data from file.  **while** (1)  {  **if** (vCount < 256)  {  dataArray[vCount] = **myData**(vCount, &dataValue);  vSum += dataArray[vCount];  **if** (dataArray[vCount] > vMax) vMax = dataArray[vCount];  **if** (dataArray[vCount] < vMin) vMin = dataArray[vCount];  vCount++;  }  **else**  {  // array full  **break**;  }  }  // Compute average only if we read at least one value  **if** (vCount > 0) {  vAvg = (uint8\_t)(vSum / vCount);  } **else** {  vAvg = 0;  }  //Sort Values  **if** (vCount > 0) {  // Sort dataArray (use 16-bit indices to avoid overflow when vCount==256)  **for**(i = 0; i < (uint16\_t)(vCount - 1); i++){  **for**(j = i + 1; j < vCount; j++){  **if** (dataArray[i] > dataArray[j]) {  uint8\_t temp = dataArray[i];  dataArray[i] = dataArray[j];  dataArray[j] = temp;  }  }  }  }  **if** (vCount > 0) {  **if** (vCount % 2 == 0) {  vMedian = (dataArray[vCount/2 - 1] + dataArray[vCount/2]) / 2;  } **else** {  vMedian = dataArray[vCount/2];  }  } **else** {  vMedian = 0;  }  //-----------Writing UART------------------//  **sprintf**(buffer, "vCount = %d \r\n", vCount);  **UART\_transmitString**(buffer);  **sprintf**(buffer, "vSum = %d \r\n", vSum);  **UART\_transmitString**(buffer);  **sprintf**(buffer, "vMax = %d \r\n", vMax);  **UART\_transmitString**(buffer);  **sprintf**(buffer, "vMin = %d \r\n", vMin);  **UART\_transmitString**(buffer);  **sprintf**(buffer, "vAvg = %d \r\n", vAvg);  **UART\_transmitString**(buffer);  **sprintf**(buffer, "vMedian = %d \r\n", vMedian);  **UART\_transmitString**(buffer);  //Turn LED ON.. done  **GPIO\_setOutputHighOnPin**(GPIO\_PORT\_P1, GPIO\_PIN0);  //Spin Loop  **while**(1){  // Nothing here.  }  } |

Terminal:

