NAME: _____

Department of Electrical and Computer Engineering

FALL 2023 COMPREHENSIVE/BREADTH EXAM

Questions 4.1	TTG Area: High Performance Embedded Computing	ECGR 4101 Adv. Embedded Systems

1) Consider an ADC with a V+ref = 3.3V, V-ref = 0v, and N=12. What is the range of voltages that Vin would have to be for the ADC output code to read "0x146"? ***Show your work below.*** Put your answer in the boxes (4 decimal points of precision). (15 points)

Vin from: [ ] V to [ ] V

NAME: _____

2) You are developing an embedded system to work with temperatures. You want to use both Celsius and Fahrenheit, and you want to have precision to tenths of degrees. However, you do not have a floating-point processor in your embedded system, and you do not have floating point libraries available in the tools.

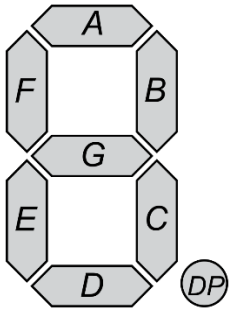
   Your goal is to write a function (called FtoC) that will convert a Fahrenheit temperature into a Celsius temperature. A helpful formula is F = (9/5 * C) + 32

   Write the syntactically and functionally correct function that will do this conversion. Follow industry accepted programming guidelines and robust design practices. (35 points)

NAME: _____

3) We have a common cathode eight segment display. We need this LED display to display only the correct alphabetic letters “A, b, c, d” when the current read ADC value is greater than zero. The ADC readings have been divided into four ranges for comparison. Assuming all the other code works correctly, debug this function code below.

   Below this function code, only write on each line that refers to the numbered line in the code if it has any coding issues, or concerns. Also write or include any syntax, or logic errors you think could prevent this code from compiling or working correctly. (50 points)



Segment A	PIN 0
Segment B	PIN 1
Segment C	PIN 2
Segment D	PIN 3
Segment E	PIN 4
Segment F	PIN 5
Segment G	PIN 6
Segment DP	PIN 7

```
// Display the correct letter according to the 12-bit ADC reading
1  int displayADCValue() {
2     if (ADCValue < 256)
3           P2OUT &= 0xF7;
4     else if (ADCValue > 255 && ADCValue < 511)
5           P2OUT &= 0x7C;
6     else if (ADCValue > 510 && ADCValue < 766)
7           P2OUT &= 0xA7;
8     else if (ADCValue > 755 || ADCValue < 1024)
9           P2OUT &= 0x5E;
10 }
```

1. _____

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

10. _____