Homework 8 Pseudo Code:

(Declare function prototypes for the following **programmer defined functions**, reference Chapter 5 lecture slides #7-#18)

Declare a function prototype to **clearSegments**. This function will take **no input**, and this function provides **no output**.

Declare a function prototype to **displayZero**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayOne**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayTwo**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayThree**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayFour**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayFive**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displaySix**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displaySeven**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayEight**. This function will take **no input and** provide **no output**.

Declare a function prototype to **displayNine**. This function will take **no input and** provide **no output**.

Declare a function prototype to **showDecimal**. This function will take **no input and** provide **no output**.

Declare a function prototype to **enableDigit**. This function will take one **integer** input, **n**, that represents one of the 4 digits on the 4-digit 7-segment display, and provides **no output**.

Declare a function prototype to **showNumber**. This function will take one **integer** input, **x**, that represents the number to show on the LED, and this function provides **no output**.

Declare a function prototype to **showRadio**. This function will take one **integer** input, **number**, and provide **no output**.

(Declare the following constant global variables, reference Chapter 13 lecture slides #6. Reference Storage Classes and Scope from Chapter 5 lecture slides #25 - #33.)

Declare a constant integer variable, **aPin**, and initialize the value to 5.

Declare a constant integer variable, **bPin**, and initialize the value to 7.

Declare a constant integer variable, **cPin**, and initialize the value to 9.

Declare a constant integer variable, **dPin**, and initialize the value to 11.

Declare a constant integer variable, **ePin**, and initialize the value to 12.

Declare a constant integer variable, **fPin**, and initialize the value to 6.

Declare a constant integer variable, **gPin**, and initialize the value to 8.

Declare a constant integer variable, **dpPin**, and initialize the value to 10.

Declare a constant integer variable, **digit1Pin**, and initialize the value to 1.

Declare a constant integer variable, **digit2Pin**, and initialize the value to 2.

Declare a constant integer variable, **digit3Pin**, and initialize the value to 3.

Declare a constant integer variable, **digit4Pin**, and initialize the value to 4.

Declare a constant integer variable, **potPin**, and initialize the value to A0.

Declare a constant integer variable, **ledPin**, and initialize the value to 13.

(Declare the following global variables. Reference Storage Classes and Scope from Chapter 5 lecture slides #25 - #33.)

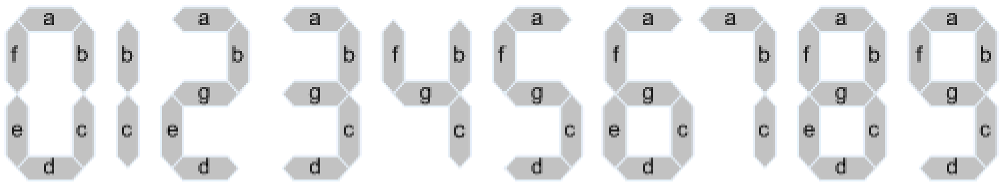
```
Declare an integer array of 6 elements, stations, and initialize the values to 889, 909, 925, 969, 1025, and 1079.

(The setup function runs once when you press reset.)
void setup()
{
  (Reference the SainSmart UNO Starter Kits Tutorials, Chapter 4: LED Blink, to set the following pins to the output mode.)
  Call the pinMode() function and set digit1Pin, digit2Pin, digit3Pin, digit4Pin, aPin, bPin, cPin, dPin, ePin, fPin, gPin, dpPin, and ledPin to OUTPUT.
}

(The loop function runs over and over again forever.)
void loop()
{
  Declare a local integer variable i.
  Declare a local unsigned long variable, val, and initialize the value to 0.
  Declare a local integer variable, freq, and initialize the value to 0.
  Declare a local integer variable, lightOn, and initialize the value to 0.
  Declare a local integer variable, timeDivisionCounter, and initialize the value to 100.

  Call the analogRead function, provide potPin as the function input, and save the analogRead function output to val.

  The value of freq can be computed by multiplying val with 205, dividing the result by 1023, and adding 875 to the result.   // this is to convert val: 0-1023 to freq: 875-1080

  For (loop i from 0 to 5)
  {
    If (the value in stations array index i is equal to the value in freq)
    {
      Increment lightOn by 1.
      Break.
    }
  }  // end of for loop
  If (lightOn is a non-zero value)
    Call the digitalWrite function and provide the ledPin and HIGH as function inputs.
  Else
    Call the digitalWrite function and provide the ledPin and LOW as function inputs.

  For (loop i from 0 to less than timeDivisionCounter)
    Call the showRadio function and provide freq as the function input.
}  // end of loop forever function
```

(The function body for the following **programmer defined functions** should be added below. Reference Chapter 5 lecture slides #7-#18)

```
void clearSegments()
{
  Call the Arduino built-in digitalWrite function to set aPin, bPin, cPin, dPin, ePin, fPin, gPin, dpPin to LOW.
}
```

```
void displayZero()
{
  Call the Arduino built-in digitalWrite function to set aPin, bPin, cPin, dPin, ePin, and fPin to HIGH.
  Set the gPin to LOW.
}
```

(Write the **displayOne**(), **displayTwo**(), …, **displayNine**() functions to set the proper pins to HIGH and LOW for each digit.)



```
Void showDecimal()
{
  Call the digitalWrite function and provide dpPin and HIGH as the function inputs.
}
```

```
void enableDigit(int n)
{
  Switch (based on the value of n)
  {
    Case 1: digitalWrite the digit1Pin to LOW, set digit2Pin, digit3Pin,digit4Pin to HIGH.
    Case 2: digitalWrite the digit2Pin to LOW, set digit1Pin, digit3Pin,digit4Pin to HIGH.
    Case 3: digitalWrite the digit3Pin to LOW, set digit1Pin, digit2Pin,digit4Pin to HIGH.
    Case 4: digitalWrite the digit4Pin to LOW, set digit1Pin, digit2Pin,digit3Pin to HIGH.
    Default: set all 4 digit pins to high.
  } // end of switch statement
} // end of enableDigit function
```

```
void showNumber(int x)
{
  Write a switch statement to call the displayZero function, displayOne function, …, displayNine function based on the value of x.  Don’t call any function in the default case.
}

void showRadio(int f)
{
  Declare an integer variable, del, and initialize the value to 55 microseconds.

  Call the enableDigit function and provide 1 as the function input.
  If (the value of f is larger than 999)
  {
    Call the showNumber function and provide (f divided by 1000) as the function input.
    Call the Arduino built-in delayMicroseconds function and provide del as the function input.
    Modify f to be the remainder of f divided by 1000.
  }
  Call the clearSegments function to turn off the lights.

  Call the enableDigit function and provide 2 as the function input.
  Call the showNumber function and provide (f divided by 100) as the function input.
  Call the Arduino built-in delayMicroseconds function and provide del as the function input.
  Modify f to be the remainder of f divided by 100.
  Call the clearSegments function to turn off the lights.

  Call the enableDigit function and provide 3 as the function input.
  Call the showNumber function and provide (f divided by 10) as the function input.
  Call the showDecimal function to show the decimal point.
  Call the Arduino built-in delayMicroseconds function and provide del as the function input.
  Call the clearSegments function to turn off the lights.

  Call the enableDigit function and provide 4 as the function input.
  Call the showNumber function and provide (the remainder of f divided by 10) as the function input.
  Call the Arduino built-in delayMicroseconds function and provide del as the function input.
  Call the clearSegments function to turn off the lights.
}  // end of showRadio function
```