Programming is to practice about how to think like computers. Please always allocate enough time to think things through.

Functions are used to reduce the repetition of the program and make the program easy to fix and maintain.

When we are going to copy and paste a code block and only change some variable names in the new block, it's time to consider to make the code block a "function". The variables that need to be changed can be the function inputs. The results can be returned back to the calling program as function outputs.

This pseudo code is written based on the use of multiple programmer defined functions.

Midterm 1 Pseudo Code:

(The following 3 programmer defined functions are used in the pseudo code.)

```
Write a computeStats function that takes 1 input and provides 1 output.
  The input is x – the long data read from the .wav file
  The output is m – the 5 statistical measurements
The computeStats function will do the following:
  Set the first element of m to the mean of data x.
  Set the second element of m to the standard deviation of data x.
  Set the third element of m to the variance of data x.
  Set the fourth element of m to the average magnitude of data x.
  Declare a variable L and set L to the length of data x.
  Declare a variable product and set product to be the multiplication between x(from element 1 to element L-1) and x(from element 2 to element L)
  Declare a variable negative. Find the values from product that is less than 0 and save the result to negative.
  Set the fifth element of m to the length of negative.
End of the computeStats function.
```

```
Write a printSignature function that takes 4 inputs and provides no output.
  The 4 printSignature function inputs are:
      • id – the file identification number
      • m – the 5 statistical measurements of a voice
      • minimum – the 5 minimum acceptable statistical measurements of a voice
      • maximum – the 5 maximum acceptable statistical measurements of a voice
  Write fprintf statement(s) to print mean, standard deviation, variance, average magnitude, and zero crossings and the minim and maximum range for each measurement to the file id.
End of the printSignature function
```

```
Write a checkStats function that takes 3 inputs and provide 1 output.
  The 3 checkStats function inputs are:
    • v – the current voice's 5 statistical measurements
    • minV – the 5 statistical measurements of the minimum acceptance criteria for a voice
```

- **maxV** – the 5 statistical measurements of the maximum acceptance criteria for a voice

The 1 **checkStats** function **output** is:

- **pass** – the number of checks that passed

```
The checkStats function will do the following:
  Declare a variable, pass, and initialize pass to 0.

  Write a for loop to loop i from 1 to 5.
    If (the value of v index i is greater than or equal to the minV index i) AND (the value of v index i is less than or equal to the maxV index i)
      Increment pass by 1.
    End of the if statement.
  End of the for loop.
End of the checkStats function.
```

(midterm1a pseudo code.)

Clear screen and clear memory.

Call the **audioread** function 3 times to read 3 voice files and save data in **y3, fs3**, **y4, fs4**, and **y5, fs5**.
Subplot and plot **y3**, **y4**, and **y5**.

Call the **computeStats** function and provide **y3** as function input. Save the function output in **m3**.
Call the **computeStats** function and provide **y4** as function input. Save the function output in **m4**.
Call the **computeStats** function and provide **y5** as function input. Save the function output in **m5**.

Compute the minimum and maximum acceptance criteria for all 5 elements from **m3**, **m4** and **m5**. Save the 5 minimum acceptance criteria in **min3**, **min4**, and **min5**. Save the 5 maximum acceptance criteria in **max3**, **max4**, and **max5**.

Call the **fopen** function to open the “voiceSignature.txt” file for writing, and save the file id in **fid1**.

Call the **fprintf** function to write the “Acceptance criteria for the word THREE” in the “voiceSignature.txt” file.
Call the **printSignature** function and provide **fid1**, **m3**, **min3**, and **max3** as the function inputs.
Call the **fprintf** function to write the “Acceptance criteria for the word FOUR ” in the “voiceSignature.txt” file.
Call the **printSignature** function and provide **fid1**, **m4**, **min4**, and **max4** as the function inputs.
Call the **fprintf** function to write the “Acceptance criteria for the word FIVE” in the “voiceSignature.txt” file.
Call the **printSignature** function and provide **fid1**, **m5**, **min5**, and **max5** as the function inputs.

Call the **fclose** function to close the “voiceSignature.txt” file.
Save all variables in **data1**.

(midterm1b pseudo code.)

Clear screen and memory.

```
Load all variables saved in data1.
Declare a variable again and initialize its value to 1.
Call the fopen function to open the “voiceCommand.dat” file for appending, and save the file id in fid2.
While the value of again is equal to 1:
  Display “voice recognition program” on screen.
  Ask the user to “input a voice file name (or press q to quit)” and save the user input in fName.
  If the value of fName is equal to ‘q’:
    Set again to 0.
    Display “goodbye” on screen.
  Else if the fName exists:
    Call the copyfile function and provide the fName and “receiveVoice.wav” as function inputs.
    Call the audioread function and provide “receiveVoice.wav” as the function input.  Save the data in r and fsr.
    Call the sound function to play the voice, provide r and fsr as the function inputs.
    In figure 1:
    Subplot and plot y3 and r.
    In figure 2:
    Subplot and plot y4 and r.
    In figure 3:
    Subplot and plot y5 and r.
    Call the computeStats function and provide r as the function input.  Save the measurements in mr.
    Call the checkStats function and provide mr, min3, max3 as function inputs.  Save results in result3.
    Call the checkStats function and provide mr, min4, max4 as function inputs.  Save results in result4.
    Call the checkStats function and provide mr, min5, max5 as function inputs.  Save results in result5.
    If the value of result3 is equal to 5:
      Display “the voice says 3” on screen.
      Print 3 in fid2.
    Else if the value of result4 is equal to 5:
      Display “the voice says 4” on screen.
      Print 4 in fid2.
    Else if the value of result5 is equal to 5:
      Display “the voice says 5” on screen.
      Print 5 in fid2.
    Else:
      Display “the voice is not 3, 4, nor 5” on screen.
    End of the inner if statement
  Else:
    Display “the file does not exsit” on screen.
  End of the if statement
End of the while loop.
Call the fclose function and provide fid2 as the function input.
```