# CSC 340 Final Software Development Project and Examination Due: Tuesday, 2 May 2017 DUE: BY 11:00 AM (FIRM)

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Due:

Your response is due by email delivery time stamped by 11:00 AM, Tuesday, 2 May 2017 (a firm deadline).

## General Instructions:

This is a software development project/take-home test and you must do your work independently. For this test you ***must employ your own implementation of matrix and vector tools implemented for your previous projects, your own implementation of complex number operations, your own implementation of fast Fourier transforms*** (Algorithms 10.3.1 ***and*** 10.7.1) ***and associated tools, to generate correlation, convolution, and spectral data.*** You may use algorithms implemented in a software tool such as Excel, MatLab, Maple or Mathematica ***only*** to construct data displays or to verify the correctness of your personal implementation of the required algorithms.

A significant component of this exercise requires the interpretation of findings as well as implementations of related algorithms (e.g., PSD, convolution, and correlation) based upon one- or two-dimensional FFTs. Give careful attention to your written responses. Give complete responses, including supporting findings and written explanations. While you must provide documentation containing evidence of your coding and your computational results, ***YOU*** MUST ASSEMBLE YOUR PROJECT RESPONSE INTO A ***SINGLE WORD DOCUMENT***. DO NOT EXPECT credit for evidence that is not included in your written response to the test questions. As always, be prepared to demonstrate functional supporting software at any point after the due date of the project.

## Signal and basic DSP Problems:

1. Common signals (24 points total):  
   For this task you will be demonstrating your understanding of common periodic functions that are synthesized from the trigonometric sine function by modifying its amplitude, frequency or phase. Since you are creating these complicated functions from simple ones, you will know what to expect from transformed representations that reveal amplitude, frequency or phase content of a periodic function.
   1. Using S terms, with S=5, 50 and 500, from each of the series below and at least 512 regularly spaced samples of x, where x ε [0, 1), generate plots of graphs of the following functions:
      1. (3 points; 1 point for each function, with all three plotted on a single chart)
      2. (3 points; 1 point for each function, with all three plotted on a single chart, not the same chart as used for the previous question)
   2. Create and plot (using the positive frequencies only) power spectral density (PSD) estimates for the functions *f500* and *g500.* (6 points)
   3. Observe that, when more terms are used for approximating the functions, the graphs appear to be approaching limit functions *fL* and *gL*.
      1. Provide a verbal description of the graphs of the two limit functions (3 points).
      2. Generate your own data to estimate the limit functions *fL* and *gL* at 512 points, and then overlay plots of their graphs onto your plot of the graphs of *f500* and *g500* (6 points)?
   4. Create and plot PSD estimates for the functions *fL* and *gL* (3 points).
2. Several in-class examples, and the functions above, employed ***sums*** of signals of the form *x(t) =* ***a*** *sin( 2π* ***f*** *(t-****c****))*, where ***a*** is the *amplitude*, ***f*** is the *frequency*, ***c*** is the *phase* shift of the signal, and *t* ε [0, 1).   
   ***Compare and contrast*** the power spectral density (PSD) estimates for two signals: x(t) = the ***sum*** of two sine functions and y(t) = the ***product*** of the same two sine functions. Assume that all sine functions have the same amplitude and the same phase shift. Also assume that f1 = 13 for the first sine function and f2 = 27 for the second, in order to generate 512 samples for *x* and *y.* (6 points total)
   1. With respect to the ***existence*** of peaks, how are the PSDs ***similar*** (3 points)?
   2. With respect to the ***locations*** of the peaks, how do the PSDs ***differ*** (3 points)?
3. The purpose of this task is for you to explore and demonstrate the effect, if any, that changes in *phase* have upon the PSD. (6 points total)
   1. Suppose that 1024 consecutive samples of a signal contain a brief pulse, represented by a single 1 and 1023 zeroes. Describe how the PSD estimate is affected by varying the phase of the pulse by varying the time at which the pulse, represented by the single 1, occurs among the 1024 samples (2 points)?
   2. Suppose that ***another*** signal consists of a pure sinusoidal tone, *h(x) = sin( 16 x)* on the interval [0, 1).
      1. How is the discrete Fourier transform affected by introducing phase shift 1>c>0, so that *h(x) = sin( 16x-c))* (2 points)?
      2. How is the PSD estimate affected by the phase shift (2 points)? Try several values for c and then report the effects, if any.
4. For this problem, use either *f50*or *g50* (***but not both***) from problem 1a. Thus, this problem description assumes that exactly 50 terms were used in forming the function. Construct ideal filters and apply them to the signal to as follows (12 points total):
   1. Use a low-pass (LP) filter that ***passes*** only the lowest five frequencies into a filtered signal.
   2. Use a high-pass (HP) filter to generate a filtered signal in which the lowest five frequencies have been ***suppressed*** and the upper 45 frequencies have been ***passed***.
   3. Use a band-pass (BP) filter to generate a filtered signal in which only the 4th through the 7th frequencies have been ***passed***.
   4. Use a notch filter to generate a filtered signal in which only the 4th through the 7th frequencies have been***suppressed***.
   5. Plot the original signal with each of the four filtered signals (8 points). How could you check your plot to know that it is correct without comparing it to the work of others?
   6. Provide a verbal description of how the original signal is affected by each of the four filtering processes (4 points):
      1. LP?
      2. HP?
      3. BP?
      4. Notch?
5. Using DSP to decode DTMF tones (a.k.a., a practical example from which you benefit many times every day). (15 points/5 points each)  
   In the directory P3data, you will find three text files, each containing integer representations of 4096 samples of a DTMF tone. The sampling frequency, fs, was 44.1 kHz, i.e., there were 44,100 samples per second. Use the data in the files to determine what DTMF tone was used to generate the tone in each file. Recall that the center frequency of bin k, is given by fk = (k\*fs)/N, and bins are counted beginning with the initial entry being bin zero, i.e., k = 0.
   1. toneA corresponds to DTMF key = \_\_\_\_\_\_\_\_\_
   2. toneB corresponds to DTMF key = \_\_\_\_\_\_\_\_\_
   3. toneC corresponds to DTMF key = \_\_\_\_\_\_\_\_\_
6. Correlation and convolution (18 points total)  
   In this section, you will use your digital signal processing tools to perform two common tasks: one, determining the range to a target by a process that is similar to what an echo-locating dolphin does physiologically; and two, smoothing a noisy signal:
   1. Assume that an acoustic pulse is transmitted in sea water and the velocity of sound in sea water in the area is approximately 1500 m/s (depending upon a myriad of factors such as depth, temperature, and salinity). Estimate the range to the primary reflector given the pulse and return signals given in “rangeTestFall2017.txt” file. Assume that the receiver is turned off for 0.01 seconds after the pulse is transmitted, 200 kHz sampling rate, and the return signal measurements were taken from the first 1024-sample window after the receiver begins listening again. How far away is the primary reflector (9 points)?
   2. Use FFT convolution to smooth the signal with a 8-point filter (p=8). On a single chart, centered on the pulse, plot the first 256 consecutive samples of the original signal and the smoothed signal (9 points).
7. Two-dimensional FFT (19 points total)  
   In this task you will be using your digital signal processing tools to locate a simple target in an uncluttered image. Specifically, you will be applying a two-dimensional fast Fourier transform to correlate a two-dimensional pulse (represented by a small, 40x90, WxH, rectangle) with the data in a large (512x512) image.  
     
   You may use the Picture class, linked to the course web site, or the Python PIL to render images for this problem.
   1. Set-up (5 points)
      1. Generate and display ***a test signal*** corresponding to a monochrome image, a 512x512 array pixel values as follows:
         1. Set all values to zero;
         2. Beginning at row 150 and column 200, change the entries to create a rectangular sub-region in the array having 120 rows and 85 columns whose values are 255
         3. On a monochrome display, the resulting image should be dark (e.g., black, corresponding to pixel values of 0) except for a 150x175 pixel rectangle that is very bright (e.g., white, corresponding to pixel values where the components r=g=b=graylevel=255).
      2. Similarly, generate and display a 40x90 test pulse using pixel intensity values = 255, and pad the remainder of a 512x512 array with zeros.
   2. Find and display the two-dimensional correlation of the signal and the test pulse (7 points).
   3. Create and render a display of the correlation magnitude, painting those locations whose correlation is within 10% of maximum with red (7 points).   
      Note: You will likely need to scale (linear of logarithmic) the correlation magnitudes. Consider using a logarithmic scaling followed by a linear scaling to the range [0, 255] for the pixel values. Also, when exploring a logarithmic scaling, remember that the correlation magnitudes may include values that are outside the domain of the log function; accordingly, you may (or will) need to translate the correlation values into the domain of the log function.