UVIC COMPUTER SCIENCE 475/575, Spring 2017
ASSIGNMENT #1

DUE January 03, 2014 (11 PM)

There are 5 regular questions and each question is worth 20 points.

IMPORTANT Each question has some additional deliverables for the graduate students taking the course. These are marked with [**CSC575**]. Undergraduate students are welcome to do them but they are optional and will not affect their grade. When a question has more than one associated number of points the first one is for CSC475 and the second for CSC575.

The assignment is worth 10% of the final grade. There is some variance in the amount of time each question probably will require. Therefore don't expect them to be equally difficult even though they are all worth the same number of points.

Some skeleton Python code is provided to help you get started with audio processing. It should be straightforward to understand but feel free to ask any questions you might have about it. Some of the questions are relatively straightforward to implement once you understand the code but some of the functionality is slightly different than what you are asked to do so make sure you modify/extend it accordingly.

You are more than welcome to do the assignment in other programming languages/environments. It will simply be a bit more work to get the basic functionality going.

Please provide your answers in a **SINGLE PDF** file. There is no need to copy the assignment specification. Also please answer the questions in order and explicitly mention if you have decided to skip a question. For questions that involve programming provide a listing of the relevant code but not all the details that are not directly relevant to the question.

Please use Connex and online submission of a **SINGLE PDF** file.

Question #1. (20 points)

Using a programming language of your choice write code to directly compute the Discrete Fourier Transform (DFT) of an input array. You should express everything directly at low level using array access, *for* loops, and arithmetic operations i.e do not use a complex number type if the language supports it, and do not use any vector/matrix multiplication facilities.

- Provide a listing of your code and a plot showing that your algorithm produces the same magnitude response as a Fast Fourier Transform routine in your language of choice that is either built in or freely available. **(5 points)**
- For testing use a linear combination of 3 sinusoids that are spaced harmonically and have different amplitudes and phases. Test a fundamental frequency that corresponds to a particular DFT bin as well as a fundamental frequency that is between DFT bins. Plot the correpsonding magnitude and phase spectrums and comment on what you observe. **(5 points)**
- Plot the basis functions i.e the sine and cosine signals for a particular DFT bin (let's say bin 4 in a 256 point DFT) **(5 points)**
- Plot the result of point-wise multiplying your input signal from the previous subquestion with the three harmonics with the two basis functions that correspond to the closest DFT bin. Also plot the result of point-wise multiplying your input signal with two basis functions that correspond to an unrelated DFT bin. What do you observe ? How do these plots connect with your understanding of the magnitude and phase for a DFT bin **(5/2 points)**

[**CSC575**] Measure how long it takes to compute 100 DFT transforms of size 256, 512, 1024, and 2048 using your direct DFT implementation and compare your measured times with the ones using some implementation of the Fast Fourier Transform in the programming language of your choice. Make some kind of plot that shows the difference in performance. **(3 points)**

Question #2. (20 points)

Using an existing FFT implementation, plot the DFT magnitude response in dB (using a DFT size of 2048) of a sine wave that is exactly equal to the center frequency of DFT bin 525. On the same plot overlap the DFT magnitude response in dB of a sine wave with frequency that would correspond to bin 600.5 (in a sense falling between the crack of two neigh- boring frequency bins). Finally on the same plot overlap the DFT magnitude rsponse in dB of the second sine wave (525.5) windowed by a hanning window of size 2048. Write 2-3 brief sentences describing and explaning the three plots and what is the effect of windowing to the magnitude response for these inputs. Make sure your plots are in dB. **(10/5 points)**

Find single note sound of an instrument (you can record yourself something using Audacity, search Freesound, etc). Pick 2048 samples of audio from the steady state portion of the sound (not the attack). Instruments such a violin, flute, organ, saxophone that can sustain long notes are better for this question. Repeat the process above with input your very short audio segment from the steady state of the instrument tone. What is the effect of windowing ? Show the corresponding plots. **(10 points)**.

[**CSC575**] Read data from a soundfile that contains a single note of a musical instrument. Plot the spectrum of 2048 samples during the steady state of the note after the initial attack with and without windowing. Plot the spectrum of 256 samples starting at the same location in time with and without windowing. Make an array of 2048 samples with the 256 samples from before followed by zeroes (zero-padding). Plot the spectrum of the 2048 zero-padded array with and without windowing. Write 2-3 brief sentences describing what you observe and show all the associated plots. **(5 points)**.

Question #3. (20 points)

Read the article "Music Retrieval: A Tutorial and Review" by Nicola Orio (available on the MIR course webpage under Course/Readings) and answer the following questions:

- Pick a piece of music that you like and try to characterize it using the time scales and music dimensions discussed in the paper. Try to be specific about the particular music piece. **(4/3 points)**
- For the same piece of music describe specifically what types of information needs the different users mentioned in the MIR overview would be interested in. **(4/3 points)**
- What is the difference between query-by-humming and query-by-example (1 paragraphs) ? **(4/3 points)**
- How is MusicXML different from MIDI (1 paragraph) ? **(4/3 points)**
- What part of the article was the most novel/interesting from your personal perspective ? Explain it to someone who has not read the article in your own words. **(4/3 points)**
- [**CSC575**] Select 3 of the cited papers in the bibliography and read them (if you can not find a specific paper online or through the UVic library choose another one). Write a short summary (1-2) paragraphs for each of them as well as a paragraph of what you thought was the most interesting/novel part. **(5 points)**.

Question #4.

- Write code that generalizes your sine wave generator to be an additive synthesis instrument i.e it should take a list of frequencies and corresponding amplitues and create a mixture sound instead of just using 3 frequencies. **(5 point)**

- Either find online or record yourself using Audacity two different sounds of the same pitch (for example violin and a flute). Examine their spectrum in Audacity and select the 10 highest peaks. Synthesize the sound using your additive synthesis instrument. Show the corresponding plots and code. Can you tell which synthesized sound is which ? What is missing ? **(5 points)**

- Read about the equal temperament tuning system and MIDI. Write code that takes a list of MIDI note numbers (not note names) and synthesizes the corresponding melody using your additive synthesis instrument. Create an audio file with a well known melody using your system and plot the corresponding spectrogram in Audacity **(10 points)**.

Question #5.

This question requires reading data from an audio file and writing data to an audio file. In almost all programming languages that I know reading data from audio files is either directly supported or there are libraries. It is also relatively straightforward to write your own input/output for uncompressed audio formats. Feel free to contact me if you have questions regarding what library to use for a particular programming environment. It also assumes that you have access to a FFT function. To better understand what is happening try the following signals: a sine wave, a simple instrument sound and a piece of music.

- Write code that reads buffers of data from the audio file, converts them to a complex spectra using the Fast Fourier Transform, converts them back to the time domain using the inverse FFT and writes the output to a file. Confirm that things are working by checking that the output file is identical to the input i.e there is no loss of information from performing the FFT and inverse FFT. **(10/5 points)**
- Convert the complex spectrum from real and imaginary coefficients to polar form i.e magnitude and phase. Replace the phases with random numbers, convert back to real and imaginary coefficients and perform the inverse FFT. What is the effect on the underlying audio ? Experiment with different sizes of window $(256, 512, 1024, ..., 32768)$. Explain what you hear based on your understanding of the DFT in 2 sentences **(5/5 points)**
- Perform a similar analysis to above but select the 4 highest peaks in the magnitude spectrum and set all the other magnitudes to zero while retaining the phases. Convert back to real and imaginary coefficients and perform the inverse FFT. What is the effect on the underlying audio ? Explain what you hear based on your understanding of the DFT in 2-3 sentences **(5/5 points)**
- [**CSC575**] The use of windowing can reduce spectral leakage as explored in Q2. Adding windowing to the processes above however has an audible effect. Overlap-add is a procedure in which the analyzed windows overlap in time in such a way as to not introduce any artifacts in the reconstruction. This is typically achieved by having windows that add up to 1 when overlapping. Implement overlap-add analysis and resynthesis and check that the resynthesis is identical to the original. Then apply the two transformations (phase randomization and

peak selection) described above. Write 2-3 sentences contrasting the results with the non-overlapping, rectangular window approach. **(5 points)**

Question #5.