Assignment 2: Discrete Fourier Transformation

Due date: 9:00am Friday November 4. No lates accepted.
Hand in: Electronic submission of all source code, data files, example output, plots. (Please follow the submission requirements in Assignment 1).

Note: Do **not** use any Fourier transformation library (Java, Python, ...). Zero marks otherwise!

Implement the basic Discrete Fourier Transformation (DFT) algorithm. You should read in an audio file in the TXT format (see assignment 1). You can presume that the file is monaural and 16-bit. Apply the DFT to the file, and generate the following tabular output in a text file (1 row per harmonic):

a) Harmonic # (including 0th)
b) The a and b coefficients for the harmonic (i.e. cosine and sine amplitudes)
c) The amplitude and phase of the harmonic

The total number of harmonics processed is up to you. It should be parameterized. Create a reasonable number, and enough to create a decent result in part D below.

Try the DFT on the following wave files:

1. Square wave (eg. from assignment 1)
2. Periodic acoustic wave (eg. Cut a piece out of the didgeridoo sample in the 4P98 wave folder; or find your own periodic wave)
3. Two seconds of recorded audio. (eg. music clip)

For each of these wave files, do the following:

A. Make a time-amplitude plot of the wave. You can make a screen shot of Audacity (or suitable editor), or plot the samples in Excel.

B. Create the DFT tables for each plot as described above, and save them as a text file.

C. Using Excel, make a histogram plot of the amplitudes of the lowest K harmonics, where you can choose a K of 20 or higher. Note that your long music file in (3) above requires a huge number of harmonics!

D. Perform a reverse transformation, by using the DFT harmonics to recreate the wave. Output the wave as a TXT audio file. Plot the wave in Audacity or Excel (hand these plots in). Verify that it sounds and looks similar to the original. You should use as many harmonics as is required to faithfully recreate the original wave file (discounting normal roundoff). The number of harmonics to use should be a user parameter.

Also, in the case of the square wave, create another version that uses only 10 harmonics. Plot this as well. Notice the difference in sound quality between it and the full-harmonic version.

Files locations:

4P98 wave folder location:	http://www.cosc.brocku.ca/Offerings/4P98/waves/
Free samples:	http://www.freesound.org/
	http://www.sampleswap.org/