Music Challenge: Part the Third
Due: Wednesday, February 27th

In this problem we take the first steps toward developing a music player that can read in the song data from a data file and not be constrained by a song that is permanently programmed in.

For this first step you should create a ASCII text file with three columns: (1) the frequency of each note in cycles per second, (2) the amplitude of each note - a number between 0 and 1 and (3) the duration of each note in seconds. For this exercise program a eight note segment of your song (therefore your file should have eight rows).

Next write a function that accepts the name of this data file as an input reads in this file and plays the eight notes through the speaker. Use the makesquare2.m function to create your series. Your program should be able to read any eight note test file that follows the above pattern as well as your data file.

Please submit electronic copies of
(1) a copy of your ASCII input file (.txt file)
(2) a copy of your program with full comments(.m file). Introductory comments must form a complete help file including instructions for creating the input file and for running the eight note player.

Grading: Grading Standard for music problem - Part the Second

	Criteria (one point each) – numbers indicate relation to the steps in table 1.7-2
1	Complete, clearly presented, follows the instructions for submittal, and was turned in on time.
Introductory comments covering:	
2	Goal of program presented (1)
3	Clear introductory comments that form a useful help file including instructions for creating an input file and for use of program (2 & 3)
4	Credit given to for all sources and assistance in completing this program
5	Comments listing all variables and their units (5)
Program logic and performance:	
6	Program can read in a ASCII file containing the frequency, amplitude and duration of eight notes.
7	.m file electronic included with essentially correct logic (5)
8	Electronic (soft) copy provided runs (5)
9	Program (soft copy) produces a song based on input file (6)
10	Output (soft copy) completely fulfills the goal of the program including no extraneous echo printing.
	Program does more than explicitly required by the assignment or some aspect of the solution is particularly well done.

Reminder on reading in ASCII files:

ASCII data files can be read into MATLAB using the *load* function. The appropriate form to use is

$$S = \text{load}(filename)$$

where *filename* is a variable containing the name of the file, including extension, as a character string (i.e. enclosed in single quotes) or as the contents of a string variable. *S* will be a two-dimensional array containing the data from the file.