

```
% Adam Siembida
% adam (dot) siembida (at) gmail (dot) com
% asiembid (at) purdue (dot) edu
% ECE301 - HW1.4
% 6/24/2009

% Reset workspace
clear all;
format compact;
clc

% Using the fact that A = 440
%           and A = (5*x)/3
x = 440*3/5;

% Three octaves of notes
uB = 15*(2*x)/8;
uA = 5*(2*x)/3;
uAsharp = uA*(2^(1/12));
uG = 3*(2*x)/2;
uF = 4*(2*x)/3;
uE = 5*(2*x)/4;
uD = 9*(2*x)/8;
uC = (2*x);

mB = 15*x/8;
mA = 5*x/3;
mAsharp = mA*(2^(1/12));
mG = 3*x/2;
mF = 4*x/3;
mE = 5*x/4;
mD = 9*x/8;
mC = x;

lB = 15*(x/2)/8;
lA = 5*(x/2)/3;
lAsharp = lA*(2^(1/12));
lG = 3*(x/2)/2;
lF = 4*(x/2)/3;
lE = 5*(x/2)/4;
lD = 9*(x/2)/8;
lC = (x/2);

%rest
r = 0;

%note durations
speed = 3; %higher number increases note duration
w = 1*speed; %whole note
h = (1/2)*speed; %half note
q = (1/4)*speed; %quarter note
e = (1/8)*speed; %eight note
```

```

s = (1/16)*speed; %sixteenth note
ds= (3/32)*speed; %dotted sixteenth note

%=====SONG DATA=====
%the first row holds the frequencies as notes
%the second row holds the duration of the notes as defined above
SongData = [mE,mE,mE,r,mC,mE,mG,lG, mC,lG,r,lE,r,lA,lB,r,lAsharp,lA;
            s, e, s,s, s, e, q, q,e+s, s,e, e,s, e, s,s,      s, e];
%           ^1st meas           ^2nd meas

SongData2 = [lG,mE,mG,mA,mF,mG,r,mE,mC,mD, lB;
            ds,ds, s, e, s, s,s, e, s, s,e+s];
%           ^3rd meas

% put first part and second part of song together
SongData = [SongData, SongData2];
%=====END SONG DATA=====

% Find out how many notes the song is
SongLength = size(SongData);
SongLength = SongLength(1,2);

% initial values of for loop
y=[];
t = 0.0001;
del = 0.0001;

% builds the notes into a single signal
for index = 1:1:SongLength
    % start and end times of the note
    start_time = t;
    end_time = start_time + SongData(2,index);

    % create time interval over which to play the note
    T = start_time:del:end_time;

    % time that the next note will start at
    t = end_time;

    % add the frequency data for the time interval
    % there arent many loops so it doesnt matter that y is not predefined
    y = [y, sin(2.*pi.*SongData(1,index).*T)];
end

% plays the song
sound(y,1/del);

```