NAME: GIDEON ADJEI

INDEX NUMBER: 3019520

DIGITAL SIGNAL PROCESSING LAB THREE REPORT

```
function xx = key2note(X, keynum, dur)
%%KEY2NOTE Producea sinusoidal waveform
corresponding to
%         given piano key number
fs = 11025;
tt = 0:(1/fs):dur
%% using the A440 as reference
freq = 440 * pow2((keynum-49)/12);

xx = real(X*exp(j*2*pi*freq*tt));
xx = xx + real(X*exp(j*2*pi*3*freq*tt));
xx = xx + real(X*exp(j*2*4*pi*freq*tt));


end
```

This is the key to note function that generates the notes based on the transcribed keys in the transcribed music. The above diagram is saved as key2note.

It takes in the amplitude of the signal, key number, duration. Added the third and fourth harmonic of of the signal to improve quality.

```
%%load bach_fugue.mat binary file
load bach_fugue.mat;

% definnig the beat rate as a global variable
%unit is beats per minute
bpm = 120;

%beats per second
bps = bpm/60;
spb = 1/bps;
spp = spb/4;   %seconds per pulse

xx = zeros(length(theVoices),
sum(scale.durations)*fs+length(scale.keys));

for i = 1:length(theVoices)
    xx(i) = zeros(1,
sum(theVoices(i).durations)*fs+length(theVoices.noteNumbers));
    for kk = 1:length(theVoices(i).noteNumbers)
      keynum = scale.theVoice.noteNumbers(kk);
      tone = key2note(100, keynum, theVoices.durations(kk));
      n2 = n1 + length(tone) - 1;
      xx([i,n1:n2]) = xx([i,n1:n2]) + tone;
      n1 = n2 + 1;
    end
end
```

The above script loops through the voices to get all the voices and then store them in an m by n vector which is then added to generate the melodies that is transcribed in the bach_fugue.mat. The file is first loaded to get the structures that contains the transcribed notes.