```

% generates 4 different song tracks from vector
% each with a different number of notes, but the types of notes
% varry with each track so that the audio of each track is the same length
% INPUT: vec: vecor of notes
% if the legtn of vec is k then
% output:
% track 1: k whole notes, one octave down
% track 2: 2k half notes, native octave
% track 3: 4k quater notes, native octave
% track 4: 8k eigth notes, one octave up
function st = song_gen(vec)
    beatsPerMin     = 200;
    timeSignature   = [4,4];
    s1   = song(           vec-7, beatsPerMin, timeSignature, 1);
    s2   = song(scale_song(vec,     2), beatsPerMin, timeSignature, 2);
    s3   = song(scale_song(vec,   4), beatsPerMin, timeSignature, 4);
    s4   = song(scale_song(vec+7,     8), beatsPerMin, timeSignature, 8);


   % making sure the arrays all have the same length (which may varry
   % slightly)
   n = max([length(s1), length(s2), length(s3), length(s4)]);
   s1(length(s1):n) = 0;
   s2(length(s2):n) = 0;
   s3(length(s3):n) = 0;
   s4(length(s4):n) = 0;

   % mix the 4 tracks by average
   st = (s1 + s2 + s3 + s4)/4;


```