

```

1  % this function returns the audio data result of playing the notes in
2  % vec interpreting each note to be a no type note and the durration of
3  % the notes is based on the time signature, bpm and type of note.
4  %INPUT:
5  % vec: is a vector of ints between 1 and 50
6  % bpm: the number of beats per minute
7  % timeSignature: a length 2 vector [N,D] where N and D are natural numbers
8  % >= 1 and D is a power of 2
9  % no: is the type of note to be played for each note in vec
10 % no is a power of 2: 1,2,4,8,16,32,...,etc
11 % E.G. song([20 22 20 25],120,[4,4],4)
12 function st = song(vec,bpm,timeSig,no)
13     N = timeSig(1);
14     D = timeSig(2);
15     bps = bpm/60;
16     sampRate = 44*1024;
17     secPerBeat = 1/bps;
18     % the length of time each note will take
19     DnoteTime = (secPerBeat)*(D/no);
20     %if no==1
21     %     DnoteTime = N*secPerBeat;
22     %end
23     st = [];
24     i = 1;
25     while (i <= length(vec))
26         % find out how many consequtive notes are the same and just play
27         % one long note
28         j = 1;
29         while ((i+j <= length(vec))&& (vec(i) == vec(i+j) ))
30             j = j+1;
31         end
32
33         % snap j down to the first power of 2 less than or equal to j
34         % this way we only play real note types
35         j = 2^floor(log(j)/log(2));
36         st = acat(st, note(vec(i),DnoteTime*j) );
37         i = i+j;
38     end
39
40     %play44(st);

```