```
 1: #include <stdio.h>
 2: #include <stdlib.h>
 3: #include <sf.h>
 4: #include <sfmisc.h>
 5:
 6: #include "fm.h"
 7: #include "cat.h"
 8:
 9: int main(void)
10: {
11:     int nchan = 1,          // number of channels (1)
12:         sr = 44100 ;        // samples per second
13:
14:     int i, nTones = 5 ;     // counter, total number of tones
15:
16:     double dB = 90 ;        // amplitude of each tone
17:
18:     double Carrier[] = {100, 200, 300, 400, 500} ;  // carrier freqs
19:     double Modulator[] = {200, 200, 200, 200, 200} ;// mod freqs
20:     double PD[] = {10, 20, 30, 40, 50} ;            // peak deviation
21:
22:     double dur = 5,         // duration of each tone
23:            startTime,       // start time for each tone
24:            totalDur ;       // total duration of the tones
25:
26:     short *tone, *output ;  // arrays for each tone, and the output
27:
28:     totalDur = 0 ;          // compute the total duration of the tones
29:     for ( i = 0 ; i < nTones ; i++ )
30:         totalDur += dur ;
31:
32:     // allocate memory for each tone, and for the output
33:     tone = (short *)Malloc(dur * sr * sizeof(short)) ;
34:     output = (short *)Malloc(totalDur * sr * sizeof(short)) ;
35:
36:     // create the tones
37:     startTime = 0 ;
38:     for ( i = 0 ; i < nTones ; i++ ) {
39:         // create one FM tone
40:         fm(tone, dur, sr, dB, Carrier[i], Modulator[i], PD[i]) ;
41:         // concatenate the sequence
42:         concatenate(output, tone, startTime, dur, sr) ;
43:         // increment the start time
44:         startTime = startTime + dur ;
45:     }
46:
47:     // save the set of tones
48:     sfsave("fmsounds.wav", output, totalDur, sr, nchan) ;
49:
50:     Free(tone) ;    // release memory
51:     Free(output) ;
52:
53:     exit(EXIT_SUCCESS) ;
54: }
```