

# Probability Software Assignment

Siddhant Godbole  
Roll no -: CS22BTECH11054

**Abstract**—In this assignment I made a Music player in C which more focuses on shuffling and plays through mpg123.

## PROCESS

- 1) I used srand to set rand and to randomise the song.  
To randomize even more I used clock ticks to get shorter timeframe - by using clock functions.
  - 2) I use two arrays to make the randomization and play .
  - 3) I take a very random number and use it to align atleast few songs by -  
-I divide the random number by total songs.  
-Remainder is the number of song to be played.  
  
-Quotient is then next divided by previous Divisor so as to get an easy random number.  
-If number becomes short a new number is taken.  
  
-The list is then played by using — mpg123  
—  
  
-Press h for help.  
  
-Number keys to seek.
- The audio is played RAW.
- 4) I have paid attention to give equal probabilities to the songs.
  - 5) Then converted the array to play the songs .

## SOME SNAPS OF MY MUSIC PLAYER



```
110 / sung$
1.mp3 2.mp3 3.mp3 4.mp3 5.mp3 6.mp3 7.mp3
8.mp3 9.mp3 10.mp3 11.mp3 12.mp3 13.mp3 14.mp3
15.mp3 16.mp3 17.mp3 18.mp3 19.mp3 20.mp3 a.out
ksuful.c playr.c

/home/siddhant/AI1110/sungs/24.mp3: File access error. (code 22)
/home/siddhant/AI1110/sungs/24.mp3aaaahasiddhant@Siddhu:~/AI1110/sungs$ .

Put number of songs -> 24
13 3 18 15 7 11 4 24 20 23 6 12 19 21 22 17 9 8 16 2 5 14 10 1
-> Put : 0 _ to exit
: 1 _ to play new list
while playing q twice to next
and hold q to exit

0 glo = nos;
1 if(nos < 100){
2     low = nos * 100;
3     upp = nos * 500000000;
4 }
5 else{
6     low = nos * nos ;
7     upp = nos * 100000000000000;
8 }
9 ent = clock ();
10 srand(time(0) + ent*nos);
11 nop = rand() % (upp - low + 1) + low ;
12
13 for( int i = 0 ; i < num ; i ++ ) songs[i] = i+1;
14
15 for( int i = 0 ; i < nos ; i++){
16     ent = clock ();
17     srand(time(0) + ent*nos);
18
19 if ( nop < nos*10 ){ nop = rand() % (upp - low + 1) + 200 ;}
20
21 ply[i] = songs[ nop % glo ];
22 mee = songs[nop % glo];
23
24 songs[ nop % glo ] = songs[glo-1];
25 songs[glo-1] = mee;
26
27 nop = (nop/nos) + ent % nos;
28
29 glo--;
30 }
31 }
32
33 for( int j = 0 ; j < num ; j++){
34     printf(" %d",ply[j]);
35 }
```

```

[s] or [ ]      interrupt/restart playback (i.e. '(un)pause')
[f]            next track
[d]            previous track
[.]            next directory (next track until directory part changes)
[']            previous directory (previous track until directory part changes)
[b]            back to beginning of track
[p]            loop around current position (don't combine with output buffer)
[.]            forward
[']            rewind
[+]            fast forward
[:]            fast rewind
[>]            fine forward
[<]            fine rewind
[+]            volume up
[-]            volume down
[u]            (un)mute volume
[r]            RVA switch
[v]            verbose switch
[l]            list current playlist, indicating current track there
[t]            display tag info (again)
[m]            print MPEG header info (again)
[c] or [C]     pitch up (small step, big step)
[x] or [X]     pitch down (small step, big step)
[w]            reset pitch to zero
[k]            print out current position in playlist and track, for the benefit of some
               external tool to store bookmarks
[h]            this help
[q]            quit

```

Fig. 1. This is how My music player looks normally