Naad-Nartan
Journal of Dance & music
UGC-CARE Listed ISSN: 2349-4654
Volume : 12, Issue-1, January 2024

Evaluation of music through C programming-Challenges and Limitations

Mrs. Kari Yamini Krishna
Research Scholar, Sri Padmavati Mahila Visvavidyalayam, Tirupati

Abstract

The development stages of music have drastically changed with technology in various areas like recording, sound generation software, and studio equipment, etc. Among all other technological aspects, basic programming languages like C, C++, and C# play a vital role in the advancement of music. This research paper mainly focuses on generating swarasthanas and the sounds of swarasthanas using C Programming Language commands. It proves that C programming language can be used to generate 12 notes of Indian music and Swarasthanas of 72 Melakarta ragas of Carnatic classical music. Using code editors like VS Code and a compatible compiler for C, these different musical sounds can be created. This process illustrates writing a C program for Mohana Raga Geetham in Carnatic Classical Music, and the Indian National Anthem. This research work also explains how to write a similar C program that displays the Swarasthanas of the chosen raga from the programmed 72 Melakartha Ragas. Challenges and limitations of the output would be discussed in the conclusion of the paper.

Keywords: C Programming, Mohana Raga, National Anthem, 72 Melakartha Ragas, Swarasthanas.

Research Paper

Introduction

Music is always accompanied by the subjects mathematics, Science and technology in every stage of its development. In Indian music the seven notes of Saptaka are further divided into 22 srutis by Bharatha in Natya Shastra by calculating frequencies of the notes. Bharatha's experiment on Veena includes both Science and Mathematics. The experiment resulted in 22 srutis. Now we are nothing in music without them. With changing scenarios and developments human beings are habituated to technology. Technology took its prior place in every subject. Technology eases the human and saves time according to the present situations. This research explores the convergence of music and programming through an in-depth analysis of a C program that employs the Windows API's 'Beep' function to generate musical melodies. The program's source code is dissected to understand its structure and logic, and its musical output is examined to assess its creative and educational potential. The findings shed light on the capabilities of code in the realm of music and its implications for both novice programmers and music enthusiasts. (Sampsel 6)

Technology and Music

With emerging technology Music (Kartz 2) also shaped its outcomes in various areas using different platforms of technology.Music technology encompasses tools and systems designed to create, preserve and modernise music according to the emerging trends.As musicians modify their body suitable for the sounds or instruments in the same way musicians also used technology to facilitate artists to use it according to their needs.

Automation of Manual Instruments using technology

Tanpuras are changed from manual to electronic. Apps like Tanpura Droid, (Application) Sruti Sadhak etc are in use by a large number of singers. The combination of Tanpura with Talas is also introduced by a few apps like Jalra, Neelambari.

Music Notations, Recordings, MP3s Online

Music Notations are also changed from printed books to online PDFs and websites.Huge collections of notations are available in online websites along with Notations, recordings etc..The most popular among such websites is (Shivkumar) launched by a software

engineer Mr.Shivkumar Kalyanaraman.The website is exclusively launched for enthusiastic music learners and listeners to know about various practical as well as theoretical aspects of Carnatic Music.

Music Teaching on various technical platforms

Learning music has become globally accessible nowadays with the help of technology. Many enthusiastic music learners are searching their favourite music institutions online and preferring to learn online. With the help of technical platforms like Zoom, Google Meet, Microsoft teams etc. many music institutions have become online nowadays.In this context I must mention here that as an IT student and MBA graduate I also tried combining music and technology by creating an online learning platform to my students. That helped them alot in terms of saving time to travel from their residence to music institution. For students of other countries like the US and UK, this platform helped in learning at their convenient time according to their daily routine. (Sadguru Music Academy)

Music and Programming Languages

Not many musicians work solely in the software industry. They are more interested in experimenting with musical concepts than in creating new applications. Libraries of reusable software modules are crucial for the majority of computer musicians since pre-made modules can make exploration easier or even provide inspiration for new musical approaches. This can occasionally prevent new languages from being adopted because they do not initially have a fully developed set of ready-made features and examples. When a musician creates a sound or "patch" (either from scratch or with the use of synthesizers and samplers), or utilizes a sequencer or compiler to organize a composition, they are said to be engaging in music programming.

Here is a list of a few notable programming Languages optimized for sound production, algorithmic composition, and sound synthesis.

- ABC notation, a language for notating music using the ASCII character set (Manaris and Brown 204)
- Bol Processor, a model of formal grammars enriched with polymeric expressions for the representation of time structures
- ChucK, strongly timed, concurrent, and on-the-fly audio programming language (Kapur 1)
- Real-time Cmix, a MUSIC-N synthesis language somewhat similar to Csound
- Cmajor, a high-performance JIT-compiled C-style language for DSP
- Common Lisp Music (CLM), a music synthesis and signal processing package in the Music V family
- Csound, a MUSIC-N synthesis language released under the LGPL with many available unit generators

Evaluation of Music through C programming

Dennis Ritchie developed the general-purpose computer language C at Bell Laboratories in 1972. Despite being an ancient language, it is extremely popular. Its widespread use is primarily due to the fact that it is a foundational language in the study of computers. The sound function produces the sound of a specified frequency. Used for adding music to a C program, we tried to use some random values in a loop, with varying delay.

One of my music students Mr. P. Kalyan who is also an engineering student of Computer science and electronics at Rajiv Gandhi University of Knowledge Technologies, Srikakulam, Andhra Pradesh, wrote code for the few basic learnings of Carnatic music.

'Geetham' in Carnatic Music

The simplest musical form in Carnatic music is called Geetham. Geetham music is fairly straightforward, and shityam is typically written to honor the gods. The pupils study Geethams as the next stage after mastering swaras (notes) through varisa practice. Geethams are the first type of music with shityam that a learner would genuinely learn, as can be said from the track. For the geetham " Varaveena" in Raga Mohanam, my student Mr.Kalyan **Harithas** has written the Code in C language.

Geetham – Varaveena: (In praise of Goddess Saraswathi)
Rāgam – **Mohanam** (Janya of 28th Melakartha Rāgam Harikāmboji) (Satyanarayana302)
Tālam – Rupakam (Chaturasra Jāthi)
Composer – Appayya Dikshitar

Ārohanam – SRGPDS (Shadjam, Chatusruti Rishabham, Antara Gāndhāram, Panchamam, Chatusruti Dhaivatham, Shadjam)

C Programme - Code:

```
#include<stdio.h>
#include<windows.h>
int main()
{/*Declaring the notes to be played in sequence using ARRAY arr */ int i,op,a
rr[]={3,3,14,14,5,4,16,16,7,6,5,5,14,5,4,3,3,12,3,4,5,6,
```

```
15,5, 4,3, 3, 12,3,3,5,4,13,4,3,3,2,11,3,3,3,
3,2,3,4,3,14,14,3,3,5,4,15,5,4,16,16,5,8,7,7,6,6,5,6,5,5,
5,4,3,4,5,6,5,4,5,4,3,3,2,1,1,2,3,3,3,3,3,2,4,3,12,1,2,1,3,
2,1,2,2,11,11};
for (i=0;;i++) /* for LOOP TO CALL EACH ELEMNT IN ARRAY*/
 {
 op=arr[i];
  switch(op) /* DECLARING THE NOTES TO BE PLAYED USING Beep FUNCTION */
 {
 case 1: Beep(261.63,500);
  break; /* Beep( Frequency of the note, Time of extension) */
 case 2: Beep(293.66,500);
 break;
 case 3: Beep(329.63,500); /* USING SWITCH CASE TO NAME EACH BEEP FUNCTION(i.e NOTE) */
 break;
 case 4:Beep(392,500);
 break;
 case 5: Beep(440,500);
 break;
 case 6: Beep(523.25,500);
 break;
 case 7: Beep(587.33,500);
 break;
 case 8: Beep(659.25,500);
 break;
 case 11: Beep(261.63,1000);
 break;
 case 12: Beep(293.66,1000);
 break;
 case 13: Beep(329.63,1000);
 break;
 case 14:Beep(392,1000);
 break;
 case 15: Beep(440,1000);
 break;
 case 16: Beep(523.25,1000);
 break;
 case 17: Beep(587.33,1000);
 break;
 case 18: Beep(659.25,1000);
 break;
 }
 }
}
```

C Programming for 72 Melakarthas of Carnatic Music

Melakartha Ragas or Janaka Ragas in carnatic music are the parent ragas that generate a huge number of Janya Ragas with the similar notes of parent raga but with a different order of Arohana and Avarohana. Melakartha Ragas are synonyms of Scale which consists of Arohana and Avarohana with different notes from shodasha swara sthanas of Carnatic music.

The C programme gives the results in terms of Arohana Avarohana of the selected number of the Raga. For example, If we select the number 29 out of 72 and run the below C program, the result shows Arohana and Avarohana of Dheera Shankara Bharanam which is 29th Melakartha of Carnatic Music.

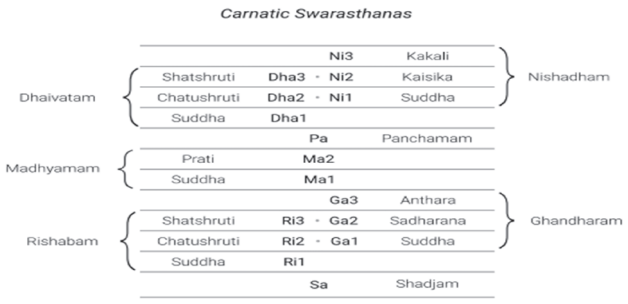

Figure 1: Shodasha Swarasthanas of Carnatic Music

C Programme - Code:

```
#include<stdio.h>
int main()
{
int num,n,rem,q,i,j,o,k,l;
printf("Enter the number of the MELAKARTHA RAAGA:"); /*TAKING INPUT */
scanf("%d",&num);
if(num>72)
        {
```

```
        printf("ENTER THE NUMBER BETWEEN 0-72");
        }
if(num>36)
        {
        n=num-36; /* If number is greater than 36 subtract it with 36 and storing value in n */
        }
else
        {
        n=num;
        }
n=n-1; /* Subtract n with 1 */

rem=n%6; /* Diving n with 6 */

q=n/6; /* Storing reminder value in rem variable and quotient value in q */

switch(q) /* Quotient value decides RI and GA placements */
        {
        case 0:
        i=1; // RI1 GA1
        j=1;
        break;
        case 1:
        i=1; // RI1 GA2
        j=2;
        break;
        case 2:
        i=1; // RI1 GA3
        j=3;
        break; // RI2 = GA1
        case 3:
        i=2; // RI2 GA2
        j=2;
        break;
        case 4:
        i=2; // RI2 GA3
        j=3;
        break; // RI3 = GA2
        case 5:
        i=3; // RI3 GA3
        j=3;
        break;
        }
if(num<=36)
        {
        o=1; /* If number is less than or equal to 36 than MA1 */
        }
else
        {
        o=2;}
switch(rem) /* Remainder value decides DHA and NI placements */
        {
        case 0:
        k=1; // DHA1 NI1
        l=1;
        break;
        case 1:
        k=1; // DHA1 NI2
        l=2;
        break;
        case 2: // DHA1 NI3
        k=1;
        l=3; // DHA2 = NI1
        break;
        case 3: // DHA2 NI2
        k=2;
        l=2;
        break;
        case 4: // DHA2 NI3
        k=2; // DHA3 = NI2
        l=3;
        break;
        case 5: // DHA3 NI3
```

```
k=3;                                    S\n",i,j,o,k,l);
l=3;                                    // PRINTING OUTPUT
break;                                  printf("S N%d D%d P M%d G%d R%d
}       printf("S R%d G%d M%d P D%d N%d S\n",l,k,o,j,i); }
```

Seventy-Two (72) Melakarta Ragas of Carnatic Music (Sharma 31)

	Suddha Madhyama M1			Prati Madhyama M2
	1 Indu Cakra			7 Rishi Cakra
1	Kanakangi (Kanakambari)	R1 G1 D1 N1	37	Salagam (Sugandini)
2	Ratnangi (Phenadhyuti)	R1 G1 D1 N2	38	Jalarnavam (Jaganmohinam)
3	Ganamurti (Ganasamavarali)	R1 G1 D1 N3	39	Jhalavarali (Dhalivarali)
4	Vanaspati (Bhanumati)	R1 G1 D2 N2	40	Navanitam (Nabhomani)
5	Manavati (Manoranjani)	R1 G1 D2 N3	41	Pavani (Kumbhini)
6	Tanarupi (Tanukirti)	R1 G1 D3 N3	42	Raghupriya (Ravikriya)
	2 Netra Cakra			**8 Vasu Cakra**
7	Senavati (Senagrani)	R1 G2 D1 N1	43	Ghavam Bodhi (Girvani)
8	Hanumatodi (Janatodi)	R1 G2 D1 N2	44	Bhavapriya (Bhavani)
9	Dhenuka(Dhunibhinnashadjam)	R1 G2 D1 N3	45	Shubhapantuvarali (Shivapantuvarali)
10	Natakapriya (Nagabharanam)	R1 G2 D2 N2	46	Shad Vidamargini (Stavarajam)
11	Kokilapriya (Kokilavan)	R1 G2 D2 N3	47	Suvarnangi (Sowviram)
12	Rupavati	R1 G2 D3 N3	48	Divyamani (Jivantika)
	3 Agni Chakra			**9 Brahma Chakra**
13	Gayakapriya (Geya Hejjajji)	R1 G3 D1 N1	49	Dhavalambari (Thavalingam)
14	Vakulabharanam (Vati Vasantha Bhairavi)	R1 G3 D1 N2	50	Namanarayani (Namadeshi)
15	MayamalavaGowla	R1 G3 D1 N3	51	Kamavardhini, Pantuvarali (Kashiramakriya)
16	Chakravakam (Toyavegavahini)	R1 G3 D2 N2	52	Ramapriya (Ramamanohari)
17	Suryakantam (Chayavati)	R1 G3 D2 N3	53	Gamanashrama (Gamakakriya)
18	Hatakambari (Jayashuddhamalavi)	R1 G3 D3 N3	54	Vishvambhari (Vamshavati)
	4 Veda Cakra			**10 Disi Cakra**
19	Jhankaradhwani (Jhankar Bhramari)	R2 G2 D1 N1	55	Shyamalangi (Shyamalan)
20	Natabhairavi (Nari Reethigowla)	R2 G2 D1 N2	56	Shanmukhapriya (Chamara)
21	Keeravani (Kiranavali)	R2 G2 D1 N3	57	Simhendra Madhyamam (Sumadyuti)
22	Kharaharapriya (Srirangam)	R2 G2 D2 N2	58	Hemavati (Deshisimharavam)
23	Gowri Manohari (Gowri Velavali)	R2 G2 D2 N3	59	Dharmavati (Dhumavati)
24	Varunapriya (Viravasantam)	R2 G2 D3 N3	60	Nitimati (Nishadam)
	5 Bana Chakra			**11 Rudra Chakra**
25	Mararanjani (Sharavati)	R2 G3 D1 N1	61	Kantamani (Kuntalam)
26	Charukeshi (Tarangini)	R2 G3 D1 N2	62	Rishabhapriya (Ratipriya)
27	Sarasangi (Surasena)	R2 G3 D1 N3	63	Latangi (Geethapriya)
28	Harikambhoji (Hari Kedaragowla)	R2 G3 D2 N2	64	Vacaspati (Bhushavali)

29	Dhira Shankarabharanam	R2 G3 D2 N3	65	Mecha Kalyani (Shantakalyani)
30	Naganandini (Nagabharanam)	R2 G3 D3 N3	66	Chitrambari (Caturangini)
	6 Rutu Chakra			12 Aditya Cakra
31	Yagapriya (Kalavati)	R3 G3 D1 N1	67	Sucharitra (Santana Manjari)
32	Ragavardhini (Ragacudamani)	R3 G3 D1 N2	68	Jyotiswarupini (Jyoti Raga)
33	Gangayabhushhani (Gangatarangini)	R3 G3 D1 N3	69	Dhatuvardani (Dhowta Panchamam)
34	Vagadeeshwari (Bhogachaya Nattai)	R3 G3 D2 N2	70	Nasikabhushani (Nasamani)
35	Shulini (Shailadeshakshhi)	R3 G3 D2 N3	71	Kosalam (Kusumakaram)
36	Chalanattai	R3 G3 D3 N3	72	Rasikapriya (Rasamanjari)

Table-1: Seventy-Two (72) Melakarta Ragas of Carnatic Music

To show the result of the C program for 72 Melakarta ragas below given screen pictures can explain clearly. In the figure-2, I have chosen the **Melakartha Raga 5** and the result in the next line is-

S R1 G1 M1 P D2 N3 S

S N3 D2 P M1 G1 R1 S

The above is the Arohana and Avarohana of Raga Manavathi which is the 5th Melakarta Raga.

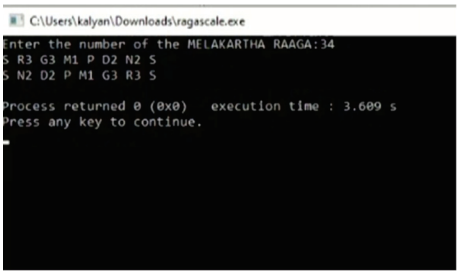

Figure-2: Arohana and Avarohana of Raga Manavathi

Similarly in the second picture figure-3 the Number of the Melakartha Raga entered is 34 which is Vagadeeshwari Raga. The result of the C program is

S R3 G3 M1 P D2 N2 S

S N2 D2 P M1 G3 R3 S

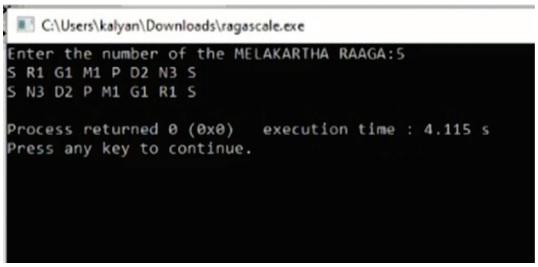

Figure 3: Arohana and Avarohana of Raga Vagadheeswari

The third picture figure-4 is the output of the C program for 69th Melakartha Raga DhatuVardhini.

S R3 G3 M2 P D1 N3 S

S N3 D1 P M2 G3 R3 S

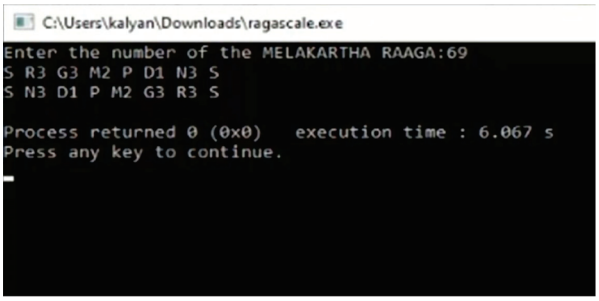

Figure-4: Arohana and Avarohana of Raga DhauVardhini

Evaluation of Music through C Programming – Code Analysis

This C program is a simple program that uses the Windows API function Beep to generate musical notes and play a melody. The program defines an array called arr that contains a sequence of integers, each of which corresponds to a specific musical note. The program then loops through this array and plays the corresponding notes using the Beep function.

Here's a breakdown of the program:

1. Includes
   - ❖ **#include<stdio.h>:** Includes the standard input/ output header for functions like printf.
   - ❖ **#include<windows.h>:** Includes the Windows API header for the **Beep function**.
   - ❖ **int main():** This is the main function of the program.
2. Declaration of an integer variable i and op. i is used as a loop counter, and op is used to store the current note to be played.
3. Definition of the arr array: This array holds the sequence of notes to be played. Each integer in the array corresponds to a specific musical note. The

numbers used here represent frequencies, with lower numbers representing lower-pitched notes and higher numbers representing higher-pitched notes.

4. for loop: This loop is an infinite loop (for(i=0;;i++)) that iterates through the arr array.
5. Inside the loop, the program assigns the current element of the arr array to the variable op.
6. switch statement: This switch statement is used to determine which note to play based on the value of op.

- ❖ Each case represents a specific note and uses the Beep function to generate the corresponding sound.
- ❖ The Beep function takes two arguments: the frequency of the note (in Hertz) and the duration of the note (in milliseconds).
- ❖ The cases are labeled with numbers, and each number corresponds to a specific musical note. The frequencies used here are standard frequencies for musical notes in the equal temperament tuning system.

7. After playing the note specified by the current value of op, the loop continues to the next iteration, and the process repeats indefinitely.
8. Since there is no exit condition for the for loop, this program will run indefinitely, playing the musical notes defined in the arr array in an endless loop.

In summary, this C program creates a musical melody by defining a sequence of notes in an array and playing those notes using the Windows Beep function. The program plays the melody continuously until it is manually terminated by the user.

Challenges and Limitations in Evaluating Music through C Programming

- ❖ Most Carnatic music is melodic. A concert may only have one supporting act for the featured performer (a vocalist or other performers like gottuvadhyam). Unlike in western music, where many instruments may be performing distinct scores, both musicians play the same melody. Thus, a composer who can readily "hear" his melody in his mind does not substantially benefit from the computer.
- ❖ Gamakas of Carnatic compositions may not be generated as it is with the C language commands. There will be a need for looping of commands to generate Gamakas as it is.
- ❖ Users of the programme must be skilled in C language or at least technically skilled. But not all the Carnatic musicians or Indian artists are technically skilled to get aware of these codes.
- ❖ Compilers of any programming language may be stuck in between the running of the programme or sometimes it may fail to run as expected.
- ❖ Commands of C programming are changing with the advancement of language. Programme must be updated and should be maintained by the user continuously.

Conclusion

With the changing trends and emerging technology, classical music is also finding its ways to improve along with the technology. This article is an example of mixing technology with Indian classical music which is a tiny part of the research about the combination of two vast subjects Music and Technology. It is always advisable to mix up Technology with Traditional music as this is the most useful method of giving the core music to the next generations.

References

Sampsel, L.J., Music research: a handbook, Oxford, 2019, p. 6

Kartz M., Music and Technology : A very Short Introduction. Oxford University Press, 2022, p. 2

Application, Tanpura Droid, Tanpura Droivd V 1.5. Swar System, 2023

Kalyanaraman, Shivkumar, Carnatic Music, Shivkumar Kalyanaraman, 30 October, 2023, http://sivkumar.org, Accessed 28/12/2023

Saduguru Music Academy, 'SMARAAGA'. Home Smaragaa, 2019, http://www.smaraaga.com, Accessed 28/12/2023

Manaris B and Brown Ar. Making music with Computers: Creative Programming in Python, CRC press, 2014. p. 204

Kapur A., Programming for Musicians and Digital Artists : Creating Music with Chuck. Manning. 2015, p.1

Nookala Chinna Satyanarayana. Raga Lakshana Sangraham. 2022, Creative Media Partners LLC. p. 302

Sarangadeva. Sangeeta Ratnakara os Sarangadeva. Adyar Library, 1945. p. 294

Sharma Akella Mallikarjuna. Akella Vaari Sookshma Sulabha Padhatulu. 5ed. Hyderabad, Akeela Foundation, 2019. p. 31

Ramachandra N.S., The Ragas of Karnatic music, Shubhi Publications, 2017, p. 118