“Dual Tone Multi-Frequency”

PROJECT REPORT

Submitted for the course:

Digital Signal Processing (ECE 2006)

By-

ADHEIK DOMINIC 16BEC0620
GEORGE SAJI 16BEC0795

Slot: L19+L20

Submitted to:

Prof.Kishore.V. Krishnan

(School of Electronics and Communication Engineering)

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ■ CHENNAI
www.vit.ac.in

October,2018

<u>CERTIFICATE</u>

This is to certify that the project work entitled **"Dual Tone Multi-Frequency**" that is being submitted by "ADHEIK DOMINIC, GEORGE SAJI" for Digital Signal Processing is a record of bonafide work done under my supervision. The contents of this project work, in full or in part, have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore

Date: October,2018

Signature of Students: Signature of Faculty:

Adheik Dominic
16BEC0620

Prof. Kishore.V. Krishnan

George Saji
16BEC0795

ACKNOWLEDGEMENTS:

Every project is successful largely due to the effort of a few people who have always given their valuable advice or lend a helping hand. We sincerely appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a success.

We, students of **Vellore Institute of Technology**, belonging to the SENSE School, Electronics and Communication (ECE) Branch, are extremely grateful to all of them who helped a great deal and gave their contribution in completing this project.

We express our gratitude to the Management and our **School Dean** for giving us an opportunity for carrying out our studies related to the project at the University.

At this juncture I feel deeply honoured in expressing my sincere thanks to **Prof. Kishore.V. Krishnan** for making the resources available at right time and providing valuable insights leading to the successful completion of our project.

I would also like to thank all the faculty members of **Vellore Institute of Technology** for their critical advice and guidance without which this project would not have been possible.

Last but not the least I place a deep sense of gratitude to our family members and our friends who have been constant source of inspiration during the preparation of this project work.

INTRODUCTION:

DTMF (dual tone multi frequency) is the signal to the phone company that you generate when you press an ordinary telephone's touch keys. In the United States and perhaps elsewhere, it's known as "Touchtone" phone (formerly a registered trademark of AT&T). DTMF has generally replaced loop disconnect ("pulse") dialing. With DTMF, each key you press on your phone generates two tones of specific frequencies. So that a voice can't imitate the tones, one tone is generated from a high-frequency group of tones and the other from a low frequency group. A DTMF signal consists of two superimposed sinusoidal waveforms whose frequencies are chosen from a set of eight standardized frequencies. These frequencies were chosen in Bell Laboratories, where DTMF signaling system were originally proposed as an alternative to pulse dialing system in telephony.
Here are the signals you send when you press your touchtone phone keys:

Digit	Low frequency	High frequency
1	697	1209 Hz
2	697	1336
3	697	1477
4	770	1209
5	770	1336
6	770	1477
7	852	1209
8	852	1336
9	852	1477
0	941	1336
*	941	1209
#	941	1477

AIM AND OBJECTIVE:

The aim of the project is to design a GUI that when given an input number from the Dialpad it represents its DTMF input waveform and filters the low and high frequency waveform and shows the respective graphical waveform.

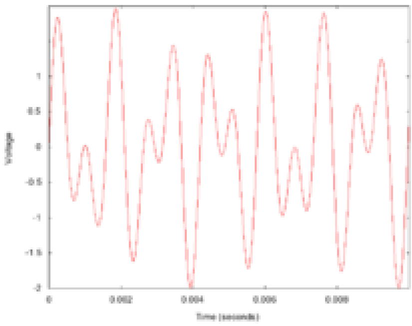
THEORY:

Dual-tone multi-frequency signalling (DTMF) is an in-band telecommunication signalling system using the voice-frequency band over telephone lines between telephone equipment and other communications devices and switching centres. DTMF was first developed in the Bell System in the United States and became known under the trademark **Touch-Tone** for use in push-button telephones supplied to telephone customers, starting in 1963. DTMF is standardized as ITU-T Recommendation Q.23. It is also known in the UK as *MF4*.

The Touch-Tone system using a telephone keypad gradually replaced the use of rotary dial and has become the industry standard for landline and mobile service. Other multi-frequency systems are used for internal signalling within the telephone network.

The keypads on telephones for the Autovon systems used all 16 DTMF signals. The red keys in the fourth column produce the A, B, C, and D DTMF events.

Keypad:



1*Combination of 1209 Hz and 697 Hz sine waves, representing DTMF "1"*

The DTMF telephone keypad is laid out as a matrix of push buttons in which each row represents the low frequency component and each column represents the high frequency component of the DTMF signal. The commonly used keypad has four rows and three columns, but a fourth column is present for some applications. Pressing a key sends a combination of the row and column frequencies. For example, the *1*key produces a superimposition of a 697 Hz low tone and a 1209 Hz high tone. Initial pushbutton designs employed levers, enabling each button to activate one row and one column contact. The tones are decoded by the switching centre to determine the keys pressed by the user.

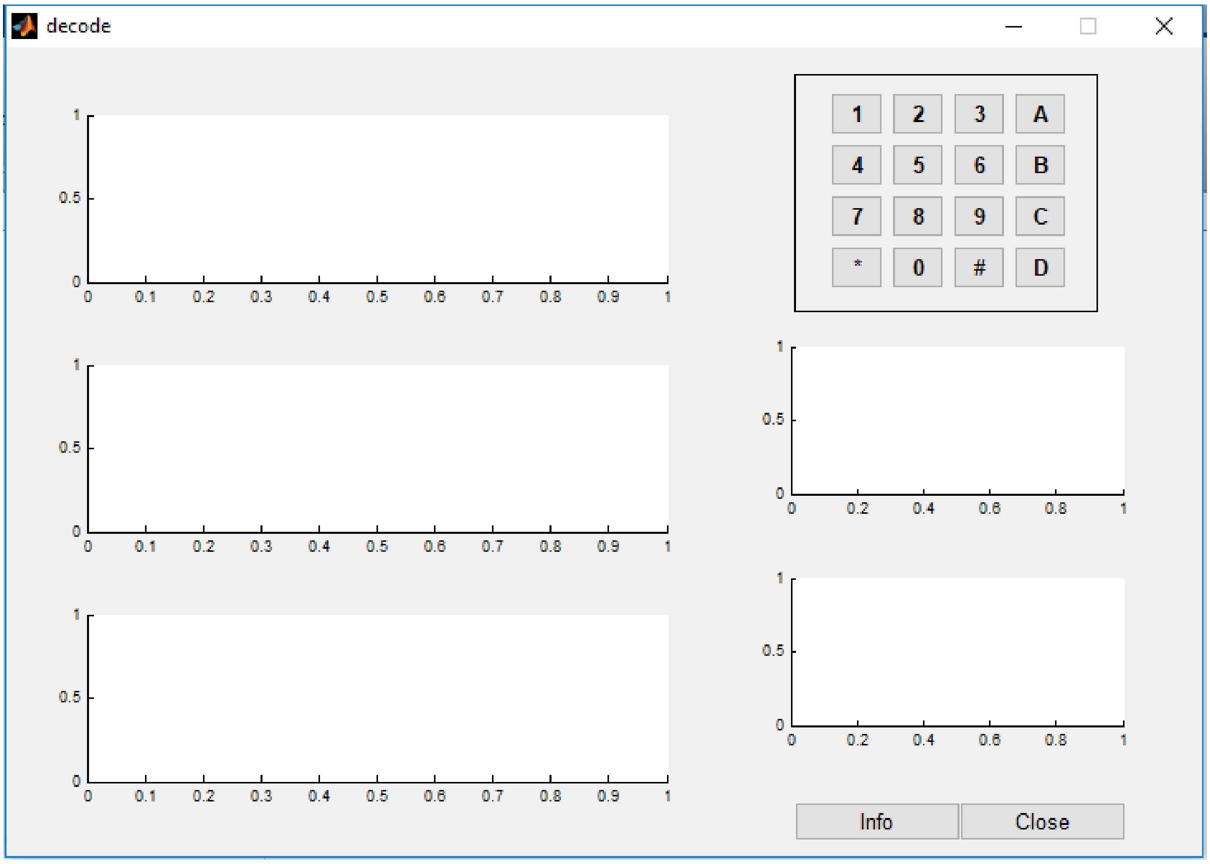
DTMF keypad frequencies (with sound clips)				
	1209 Hz	**1336 Hz**	**1477 Hz**	**1633 Hz**
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Decoding:

DTMF was originally decoded by tuned filter banks. By the end of the 20th century, digital signal processing became the predominant technology for decoding. DTMF decoding algorithms typically use the Goertzel algorithm. As DTMF signalling is often transmitted in-band with voice or other audio signals present simultaneously, the DTMF signal definition includes strict limits for timing (minimum duration and interdigit spacing), frequency deviations, harmonics, and amplitude relation of the two components with respect to each other (*twist*).

ALOGRITHM:

- That’s the GUI used for DTMF Decoder using MATLAB

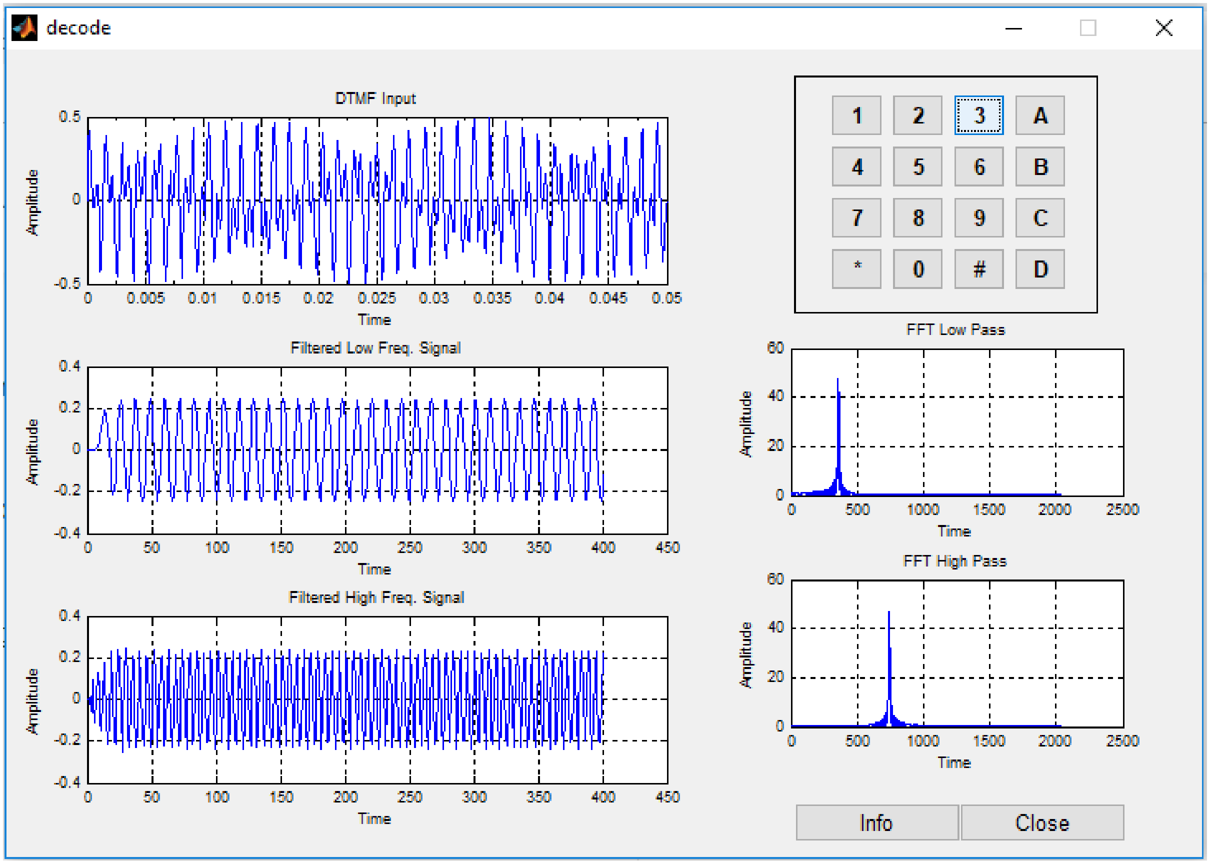


- You can see a keypad is shown in the above GUI, now I have assigned a specific tune to each of these buttons and the code for assigning this tune is as follows:

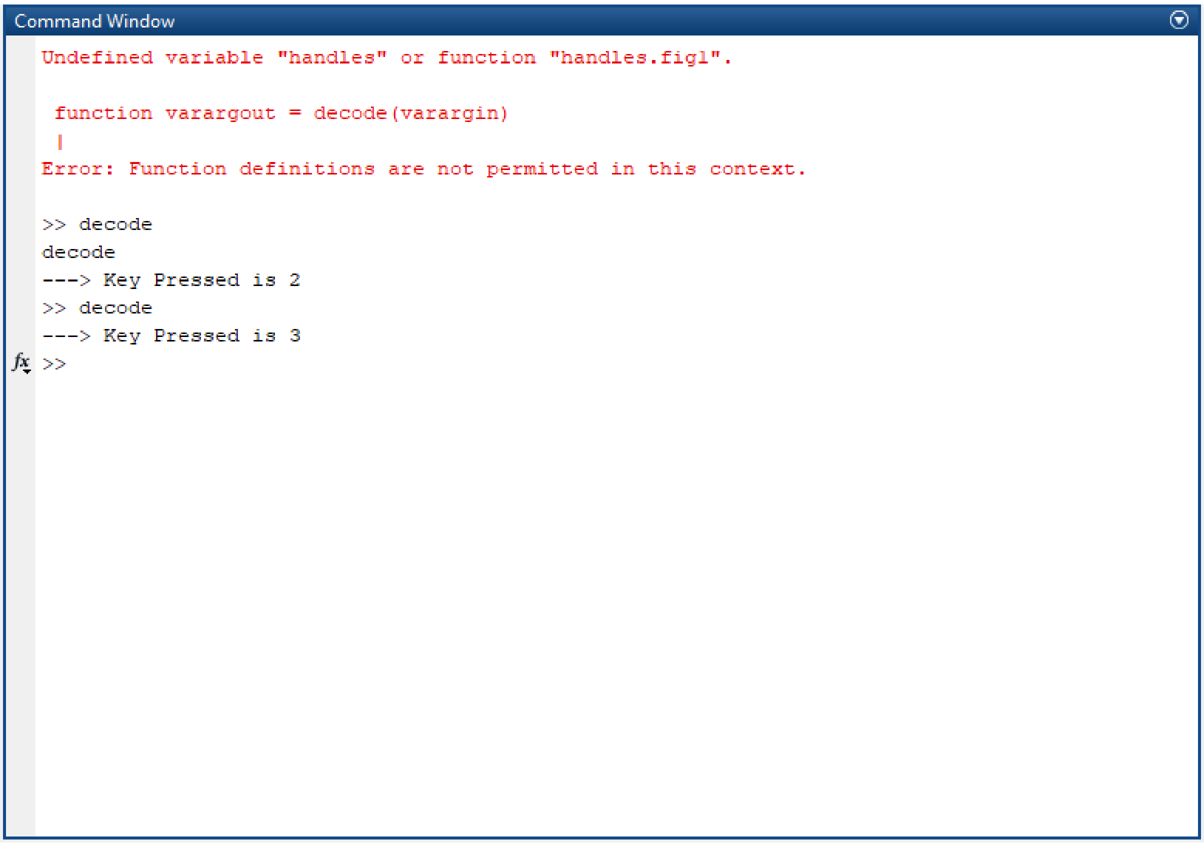
```
t=[0:0.000125:.05];
fs=8000;
f1=770;f2=1477;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
```

- In the above code we have generated a sine wave and then created a sound using that sine wave.
- So, we have such sounds assigned to each of these buttons.
- Now once button is pressed, the respective sound will be activated and right after that sound, we have added a subroutine for decoding that sound.
- This subroutine is placed in a separate file named as subdecode.m.
- This subdecode.m is responsible for DTMF decoding.

- Now we can see in this code we are applying FFT on each of these sound signals and then comparing them to get our required button press.
- Now, when we press any of these buttons then the GUI will look something as shown in below figure:



- we can see in the above figure that first graph is showing the DTMF input, which is actual signal which we have converted to sound on button press.
- The second graph is showing the Filtered Low Frequency Signal while the third one is showing the Filtered High Frequency Signal.
- The two graphs on the right side are showing the Amplitude of FFT Low Pass and FFT High Pass.
- Now if you have a look at the Command window of MATLAB then it will give you the button pressed as shown in below figure:



- These are the buttons which we have pressed while testing it and it has given me each time which button is pressed.

CODE:

DECODE.M

```
function varargout = decode(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @decode_OpeningFcn, ...
                   'gui_OutputFcn',  @decode_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function decode_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = decode_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;


function b1_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=697;f2=1209;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;
sound(y,fs)
subdecode;
function b2_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=697;f2=1336;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function A_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=697;f2=1663;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
```

```
subdecode;
function b3_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=697;f2=1447;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b4_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=770;f2=1209;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b5_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=770;f2=1336;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function B_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=770;f2=1633;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b6_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=770;f2=1477;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b7_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=852;f2=1209;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b8_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=852;f2=1336;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function C_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
```

```
f1=852;f2=1633;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b9_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=852;f2=1477;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function ba_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=941;f2=1209;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function b0_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=941;f2=1336;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function D_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=941;f2=1633;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs)
subdecode;
function bn_Callback(hObject, eventdata, handles)
t=[0:0.000125:.05];
fs=8000;
f1=941;f2=1477;
y1=.25*sin(2*pi*f1*t);
y2=.25*sin(2*pi*f2*t);
y=y1+y2;sound(y,fs);
subdecode;

function info_Callback(hObject, eventdata, handles)
function close_Callback(hObject, eventdata, handles)
close;
```

SUBDECODE.M

```
axes(handles.fig1);
plot(t,y);
set(handles.fig1,'XMinorTick','on');
title('DTMF Input');xlabel('Time');
ylabel('Amplitude');grid;

rmain=2048*2;rmag=1024*2;
cn=9;cr=0.5;
cl=.25;ch=.28;
[b,a]=cheby1(cn,cr,cl);
yfilt1=filter(b,a,y);
h2=fft(yfilt1,rmain);
hmag2=abs(h2(1:rmag));
[b1,a1]=cheby1(cn,cr,ch,'high');
yfilt2=filter(b1,a1,y);
h3=fft(yfilt2,rmain);
hmag3=abs(h3(1:rmag));

axes(handles.fig2);
plot(yfilt1);grid;
title('Filtered Low Freq. Signal');
xlabel('Time');ylabel('Amplitude');

axes(handles.fig3);
plot(yfilt2);grid;
title('Filtered High Freq. Signal');
xlabel('Time');ylabel('Amplitude');

hlow=fft(yfilt1,rmain);
hmaglow=abs(hlow);
axes(handles.fig4);
plot(hmaglow(1:rmag));
title('FFT Low Pass');grid;
xlabel('Time');ylabel('Amplitude');

hhigh=fft(yfilt2,rmain);
hmaghigh=abs(hhigh);
axes(handles.fig5);
plot(hmaghigh(1:rmag));
title('FFT High Pass');grid;
xlabel('Time');ylabel('Amplitude');

m=max(abs(hmag2));n=max(abs(hmag3));
o=find(m==hmag2);p=find(n==hmag3);
j=((o-1)*fs)/rmain;
k=((p-1)*fs)/rmain;

if j<=732.59 & k<=1270.91;
  disp('---> Key Pressed is 1');
  elseif j<=732.59 & k<=1404.73;
    disp('---> Key Pressed is 2');
  elseif j<=732.59 & k<=1553.04;
    disp('---> Key Pressed is 3');
  elseif j<=732.59 & k>1553.05;
```

```
    disp('---> Key Pressed is A');
  elseif j<=809.96 & k<=1270.91;
    disp('---> Key Pressed is 4');
  elseif j<=809.96 & k<=1404.73;
    disp('---> Key Pressed is 5');
  elseif j<=809.96 & k<=1553.04;
    disp('---> Key Pressed is 6');
  elseif j<=809.96 & k>1553.05;
    disp('---> Key Pressed is B');
  elseif j<=895.39 & k<=1270.91;
    disp('---> Key Pressed is 7');
  elseif j<=895.39 & k<=1404.73;
    disp('---> Key Pressed is 8');
  elseif j<=895.39 & k<=1553.04;
    disp('---> Key Pressed is 9');
  elseif j<=895.39 & k>1553.05;
    disp('---> Key Pressed is C');
  elseif j>895.40 & k<=1270.91;
    disp('---> Key Pressed is *');
  elseif j>895.40 & k<=1404.73;
    disp('---> Key Pressed is 0');
  elseif j>895.40 & k<=1553.04;
    disp('---> Key Pressed is #');
  elseif j>895.40 & k>1553.05;
    disp('---> Key Pressed is D');
end
```

Output:

Output on input number 2: