***5.1 Function to create a signal with Harmonics :-***

function [y,fs,duration] = harmonics(duration,f,fs,n)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

t=0:1/fs:duration;

%time sampling with 1/fs in a given time duration

harmamps=rand(1,n);

harmphase=rand(1,n);

%calculating 'n' random inputs for harmonic phase and amplitude

y=zeros(size(t));

for k = 1:n

y = y+harmamps(k)\*sin(2\*pi\*k\*f\*t+harmphase(k));

%Loop over Harmonics adding weighted versions to y

end

y = y/max(y(:))\*0.95;

%Normalize maximum amplitude to 0.95 so that sound (y,fs) does not get

%distorted

end

***Script to plot the above calculated signal :-***

sound(y,fs); %play the sound of harmonic sinusoidal signa

plot(0:1/fs:duration,y);%plotting the Harmonic Sinusoidal Signal

title("Calculating the Signal using the Sinusoidal harmonics");

ylabel("y(t)")

xlabel("time")

***5.2 Function to create a Signal Envelope :-***

function env = envelope(fs,a,d,s,dur,r)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

%In each phase of the signal determine the corresponding piece of time

%vector and envelope

t=0:1/fs:a;

env = (1/a)\*t; %Attack : Signal linearly decreases from 0 to 1 in seconds

tdelay = (a+1/fs):1/fs:a+d;

envdelay = 1 + (tdelay-a)\*(s-1)/d; %Decay: Signal linearly decreases from 1 to s in d seconds

t = [t,tdelay];

env = [env,envdelay];

tsustain = a+d+1/fs:1/fs:a+d+dur;

envsustain = zeros(size(tsustain)); %Sustain : Signal stays at s for dur seconds

envsustain = envsustain+s;

t = [t,tsustain];

env = [env,envsustain];

trelease = a+d+dur+1/fs:1/fs:a+d+dur+r;

envrelease = (-1\*s/r)\*(trelease-a-d-dur)+s; %Release : Signal linearly decreases from s to 0 in r seconds

t = [t,trelease];

env = [env,envrelease];

end

***Script to plot the above calculated Harmonic signal as in 5.1 along with its enveloped version as in 5.2 :-***

sound(y,fs);%play the sound of harmonic sinusoidal signal

subplot(2,1,1)

plot(0:1/fs:duration,y);%plotting the Harmonic Sinusoidal Signal

title("Calculating the Signal using the Sinusoidal harmonics");

ylabel("Sinusoidal Harmonic Signal amplitude");

xlabel("time");

pause(3); %delay the sound of the output to hear the differencs between harmonic sinusoidal signal and the enveloped harmonic sinusoidal signal

sound(y.\*env,fs);%play the sound of enveloped harmonic sinusoidal signal

subplot(2,1,2)

plot(0:1/fs:duration,y.\*env);%ploting the enveloped harmonic sinusoidal signal

title("Enveloping the above Harmonic Sinusoidal Signal")

xlabel("time");

ylabel("Enveloped Harmonic Sinusoidal Signal amplitude")

***5.3 Function to generate, hear and plot a simple music synthesizer :-***

function z = synthesizer(notes,fs,a,d,s,dur,r)

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

z=[]; %Initialize output as empty

durs=a+d+dur+r; %time duration of each note

p=length(notes);%calculating no. of inputs

t\_syn=0:1/fs:p\*durs+(p-1)/fs;

%loop over the notes

for k=1:length(notes)

env = envelope(fs,a,d,s,dur,r); %Compute the time vector and ADSR envelope for this note

h=harmonics(a+d+dur+r,notes(k),fs,100);%Compute the sum of harmonics for this note

n = h.\*env;%modulate the sum of harmonics with the envelope

z = [z,n];% add the note to the sequence

end

sound(z,fs); %To play the sound

plot(t\_syn,z);%plotting the different notes with their respective duartions

title("Musical Synthesizer")

xlabel("time");

ylabel("Output Signal taking different notes and their harmonics");

end