clc;

clear all;

close all;

n = 0:.005:1;

amp = 3;

xn = amp \* sin (2 \* pi \* 5 \* n);

subplot(221)

stem(n,xn)

title('continuous value discrete time signal')

bit = 2;

level = 2^bit;

r = amp-(-amp);

step\_size = r/level;

level\_values = -amp:step\_size:amp

no\_sample = length(n)

xq = zeros(1,no\_sample)

for i=1:1:no\_sample % for loop to run in values of xn

for j=-amp:step\_size:amp-step\_size % for loop to run among the levels

if xn(i) >= j && xn(i) <= j+step\_size % to compare value of xn within the level

sub1 = xn(i) - j;

sub2 = j+step\_size - xn(i);

if sub1<= sub2 % to quantize nearest value

xq(i) = j;

elseif sub1 > sub2

xq(i) = j+step\_size;

end

end

end

end

subplot(222)

stem(n,xq,'c')

title('quantized 2 bit signal')

bit = 3;

level = 2^bit;

r = amp-(-amp);

step\_size = r/level;

level\_values = -amp:step\_size:amp

no\_sample = length(n)

for i=1:1:no\_sample

for j=-amp:step\_size:amp-step\_size

if xn(i) >= j && xn(i) <= j+step\_size

sub1 = xn(i) - j;

sub2 = j+step\_size - xn(i);

if sub1<= sub2

xq(i) = j;

elseif sub1 > sub2

xq(i) = j+step\_size;

end

end

end

end

subplot(223)

stem(n,xq,'g')

title('quantized 3 bit signal')

bit = 8;

level = 2^bit;

r = amp-(-amp);

step\_size = r/level;

level\_values = -amp:step\_size:amp

no\_sample = length(n)

for i=1:1:no\_sample

for j=-amp:step\_size:amp-step\_size

if xn(i) >= j && xn(i) <= j+step\_size

sub1 = xn(i) - j;

sub2 = j+step\_size - xn(i);

if sub1<= sub2

xq(i) = j;

elseif sub1 > sub2

xq(i) = j+step\_size;

end

end

end

end

subplot(224)

stem(n,xq,'r')

title('quantized 4 bit signal')