Lab-6

Name:D.Manogna

Roll No:2022102021

Name:SriVarshitha

Roll No:2022102030

1)

a)

Here p[n]=cos((2*pi*n*fo)/fs)

The DTFT of p[n] consists of impulses at frequencies +2*pi*fo/fs, -2*fo*pi/fs (due to the cos terms) , these impulses are centered around the origin (w=0) . The magnitudes of the impulses are determined by the amplitude of the cos wave.

Lab-6

1.

a) Given

$$P[n]=\cos\left(\frac{2\pi n f_0}{f_s}\right)$$

DTFT of $P[n] = P(e^{jw})$

Let $\frac{2\pi f_0}{f_s} = x$.

$$P(e^{jw}) = \sum_{n=-\infty}^{\infty} p[n]\, e^{-jwn}$$

$$= \sum_{n=-\infty}^{\infty} \cos\left(\frac{2\pi n f_0}{f_s}\right) e^{-jwn}$$

$$= \sum_{n=-\infty}^{\infty} \cos(nx)\, e^{-jwn} = \sum_{n=-\infty}^{\infty} \left(\frac{e^{njx} + e^{-(njx)}}{2}\right) e^{-jwn}$$

$$= \frac{1}{2}\left(\sum_{n=-\infty}^{\infty} e^{j(x-w)n} + \sum_{n=-\infty}^{\infty} e^{-j(x+w)n}\right)$$

$$= \frac{1}{2}\left(\sum_{n=0}^{\infty} e^{j(x-w)n} + \sum_{n=1}^{\infty} e^{j(w-x)n}\right) + \frac{1}{2}\left(\sum_{n=0}^{\infty} e^{-j(x+w)n} + \sum_{n=1}^{\infty} e^{j(x+w)n}\right)$$

$$= \frac{1}{2}\left(\frac{1}{1-e^{j(x-w)}} + \frac{e^{j(w-x)}}{1-e^{j(w-x)}}\right) + \frac{1}{2}\left(\frac{1}{1-e^{-j(w+x)}} + \frac{e^{j(w+x)}}{1-e^{j(w+x)}}\right)$$

$$= \frac{1}{2}\left(\frac{1+e^{j(w-x)}}{1-e^{j(w-x)}}\right) + \frac{1}{2}\left(\frac{\cancel{-e^{j(w+x)}}}{\cancel{1-e^{j(w+x)}}} + \frac{\cancel{e^{j(w+x)}}}{\cancel{1-e^{j(w+x)}}}\right)$$

$$= \frac{1}{2}\left(\frac{1+e^{j(w-x)}}{1-e^{j(w-x)}}\right) = \frac{1}{2}\left(\frac{e^{jx}+e^{jw}}{e^{jx}-e^{jw}}\right)$$

$$\boxed{P(e^{jw}) = \frac{1}{2}\left(\frac{e^{\frac{2\pi j f_0}{f_s}} + e^{jw}}{e^{\frac{2\pi j f_0}{f_s}} - e^{-jw}}\right)}$$

*Here if we equate the P(e^jw) to zero then we get the w=2*pi*fo/fs.

b)Relationship between impulses:

In the DTFT of p[n], the impulses at +2*pi*fo/fs, -2*fo*pi/fs are symmetric with respect to the origin this symmetry arises from the real values nature of the cos function . Mathematically , it means that P(e^jw) is conjugate symmetric , i.e., **P(e^jw)=P*(e^(-jw))

Where P*(e^(-jw)) is the conjugate complex of the P(e^jw) . This implies that the magnitude spectrum is even , and the phase spectrum is odd.

Here w[n]=1 $(0 \le n \le L-1)$

w[n]=0 for all other values of n

x[n]=p[n].w[n]

c)

c) Given

$x[n] = p[n] \times w[n]$ where

$p[n] = \cos\left(\frac{2n\pi f_0}{f_s}\right)$ and

$w[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{o.w} \end{cases}$

DTFT of $x[n]$ is $X(e^{jw})$

$X(e^{jw}) = \text{DTFT}\left(p[n] \times w[n]\right)$

$$\boxed{X(e^{jw}) = P(e^{jw}) * W(e^{jw})}$$

$\Rightarrow X(e^{jw}) = P(e^{jw}) * W(e^{jw})$

$\Rightarrow X(w) = P(w) * W(w)$

$$= \sum_{K=-\infty}^{\infty} P(K) \cdot W(w-K)$$

$$= \sum_{K=-\infty}^{\infty} \frac{1}{2}\left(\frac{e^{\frac{2\pi j f_0}{f_s}} + e^{jK}}{e^{\frac{2\pi j f_0}{f_s}} - e^{jK}}\right)\left(\frac{1 - e^{-jL(w-K)}}{1 - e^{-j(w-K)}}\right)$$

$$P(e^{jw}) = \frac{1}{2}\left(\frac{e^{\frac{2\pi j f_0}{f_s}} + e^{jw}}{e^{\frac{2\pi j f_0}{f_s}} - e^{jw}}\right)$$

$$W(e^{jw}) = \sum_{n=-\infty}^{\infty} w[n] e^{-jwn}$$

$$= \sum_{n=0}^{L-1} 1 \cdot e^{-jwn}$$

$$= \frac{1 - e^{-jwL}}{1 - e^{-jw}}$$

$$\boxed{X(w) = \sum_{K=-\infty}^{\infty} \frac{1}{2}\left(\frac{e^{\frac{2\pi j f_0}{f_s}} + e^{jK}}{e^{\frac{2\pi j f_0}{f_s}} - e^{jK}}\right)\left(\frac{1 - e^{-jL(w-K)}}{1 - e^{-j(w-K)}}\right)}$$

If we window the signal p[n] using a rectangular window w[n] , given as

x[n]=p[n].w[n]

here we are using the multiplication property of DTFT:

X(e^jw)=P(e^jw)*W(e^jw)

Where

p[n]=cos((2*pi*n*fo)/fs)

and

W(e^jw) is the DTFT of the rectangular window w[n] .

The effect of the window on the spectrum is to convolve P(e^jw) with W(e^jw), resulting in a change in the spectral shape and sidelobe levels . The mainlobe width will also be affected.

d)

here fo=12Hz

fs=64Hz

L=16

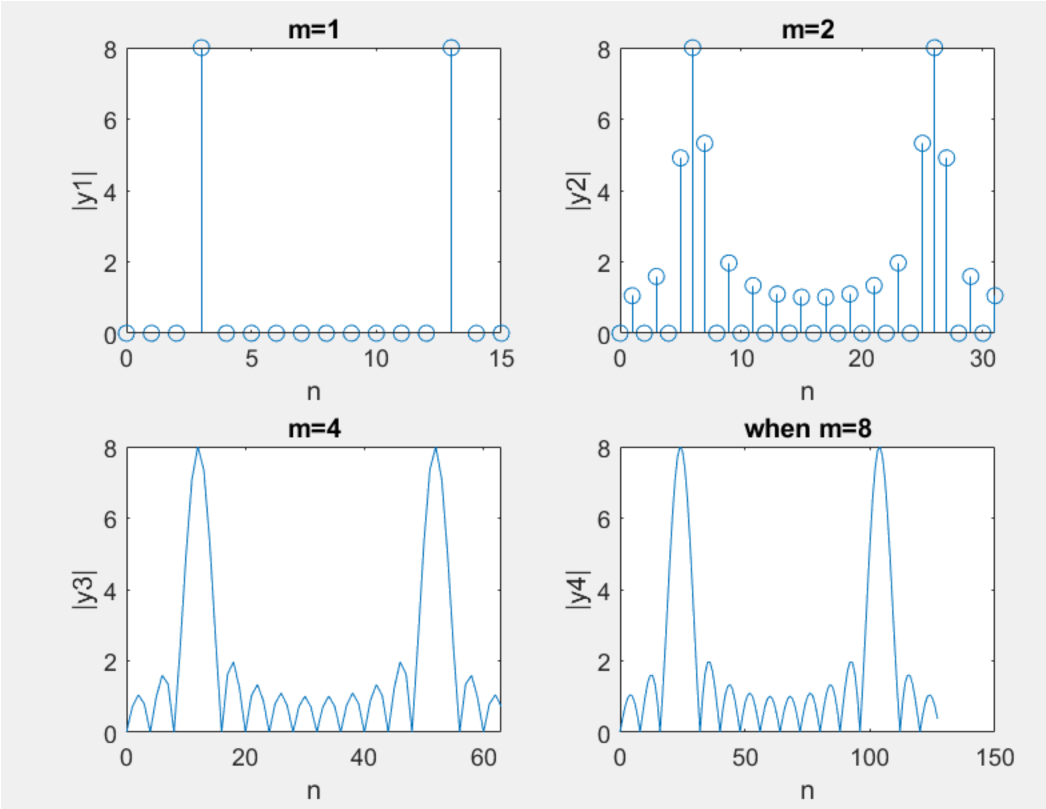
We compute the DFT X[k] for the different values of N=m*L where m={1,2,4,8}

The fft automatically performs zero padding if L<N so here we are using the fft command .

CODE:

```
fo=12;
fs=64;
L=16;
n1=0:15;
n2=0:31;
n3=0:63;
n4=0:127;
pn1=cos((2*pi*fo*n1)/fs);
pn2=cos((2*pi*fo*n2)/fs);
pn3=cos((2*pi*fo*n3)/fs);
pn4=cos((2*pi*fo*n4)/fs);
wn1=(0<=n1 & n1<=15);
wn2=(0<=n2 & n2<=15);
wn3=(0<=n3 & n3<=15);
wn4=(0<=n4 & n4<=15);
xn1=pn1.*wn1;
xn2=pn2.*wn2;
xn3=pn3.*wn3;
xn4=pn4.*wn4;
y1=fft(xn1,16);
y2=fft(xn2,32);
y3=fft(xn3,64);
y4=fft(xn4,128);
figure
subplot(2,2,1)
stem(n1,abs(y1));
title('m=1')
xlabel('n');
ylabel('|y1|');
subplot(2,2,2)
stem(n2,abs(y2));
title('m=2')
xlabel('n');
ylabel('|y2|');
subplot(2,2,3)
plot(n3,abs(y3));
title('m=4')
xlabel('n');
ylabel('|y3|');
subplot(2,2,4)
plot(n4,abs(y4));
title('when m=8')
xlabel('n');
ylabel('|y4|')
```

Plot:



Mainlobe width:

Compare the width of the mainlobe in the DFT plots here and with the part c .The main lobe width depends on the DFT length N.

Spectral Leakage:

Spectral leakage occurs when energy from one frequency component spreads to neighbouring bins .

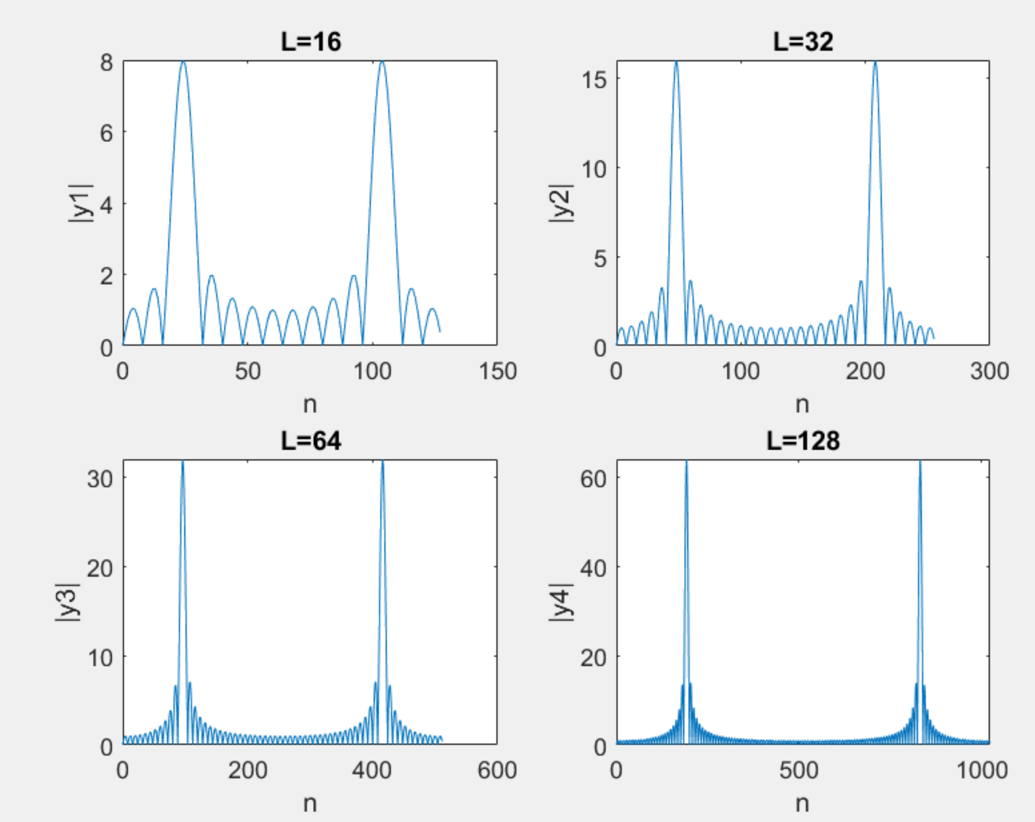
So the plots are not consistent with the part c answers (we can say this from the Mainlobe width and spectral leakage concepts).

e)

CODE:

```
fo=12;
fs=64;
n1=0:127;
n2=0:255;
n3=0:511;
n4=0:1023;
pn1=cos((2*pi*fo*n1)/fs);
pn2=cos((2*pi*fo*n2)/fs);
pn3=cos((2*pi*fo*n3)/fs);
pn4=cos((2*pi*fo*n4)/fs);
wn1=(0<=n1 & n1<=15);
wn2=(0<=n2 & n2<=31);
wn3=(0<=n3 & n3<=63);
wn4=(0<=n4 & n4<=127);
xn1=pn1.*wn1;
xn2=pn2.*wn2;
xn3=pn3.*wn3;
xn4=pn4.*wn4;
y1=fft(xn1,128);
y2=fft(xn2,256);
y3=fft(xn3,512);
y4=fft(xn4,1024);
figure
subplot(2,2,1)
plot(n1,abs(y1));
title('L=16')
xlabel('n');
ylabel('|y1|');
subplot(2,2,2)
plot(n2,abs(y2));
title('L=32')
xlabel('n');
ylabel('|y2|');
subplot(2,2,3)
plot(n3,abs(y3));
title('L=64')
xlabel('n');
ylabel('|y3|');
subplot(2,2,4)
plot(n4,abs(y4));
title('L=128')
xlabel('n');
ylabel('|y4|');
```

PLOT:



Observation:

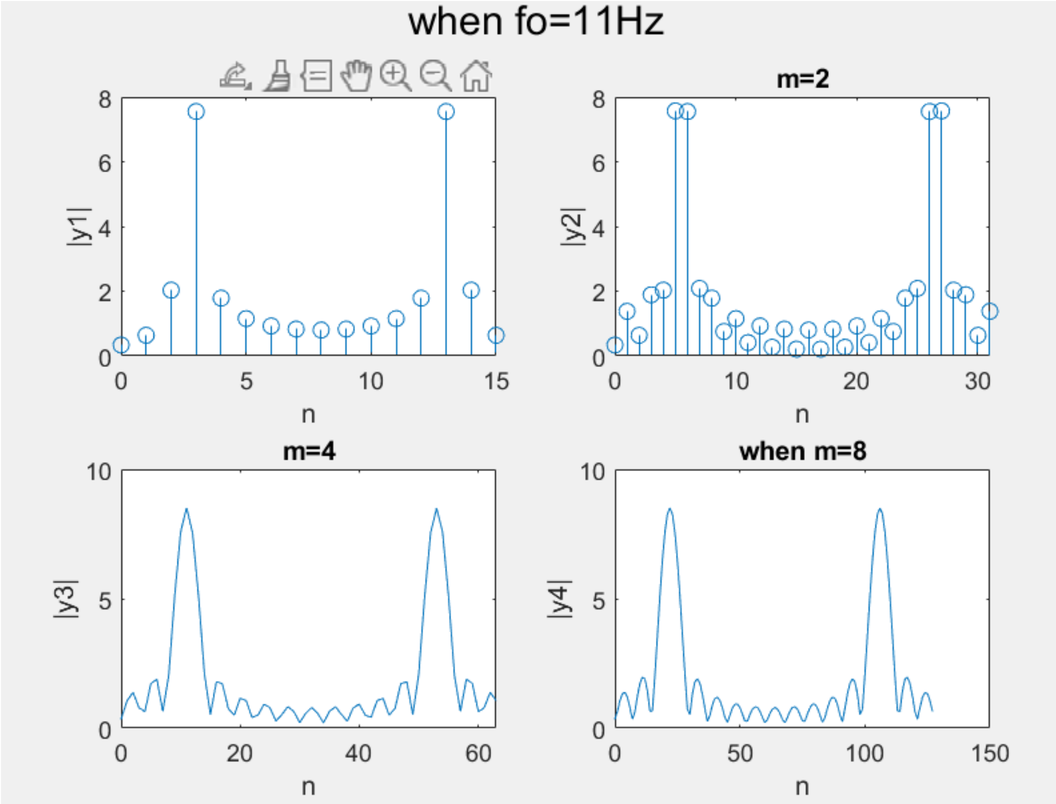
From the plots we can say that the Mainlobe width decreases and leading to better Frequency Resolution with the increase of L.

f)

CODE:

```
fo=11;
fs=64;
L=16;
n1=0:15;
n2=0:31;
n3=0:63;
n4=0:127;
pn1=cos((2*pi*fo*n1)/fs);
pn2=cos((2*pi*fo*n2)/fs);
pn3=cos((2*pi*fo*n3)/fs);
pn4=cos((2*pi*fo*n4)/fs);
wn1=(0<=n1 & n1<=15);
wn2=(0<=n2 & n2<=15);
wn3=(0<=n3 & n3<=15);
wn4=(0<=n4 & n4<=15);
xn1=pn1.*wn1;
xn2=pn2.*wn2;
xn3=pn3.*wn3;
xn4=pn4.*wn4;
y1=fft(xn1,16);
y2=fft(xn2,32);
y3=fft(xn3,64);
y4=fft(xn4,128);
figure
subplot(2,2,1)
stem(n1,abs(y1));
title('m=1')
xlabel('n');
ylabel('|y1|');
subplot(2,2,2)
stem(n2,abs(y2));
title('m=2')
xlabel('n');
ylabel('|y2|');
subplot(2,2,3)
plot(n3,abs(y3));
title('m=4')
xlabel('n');
ylabel('|y3|');
```

PLOT:

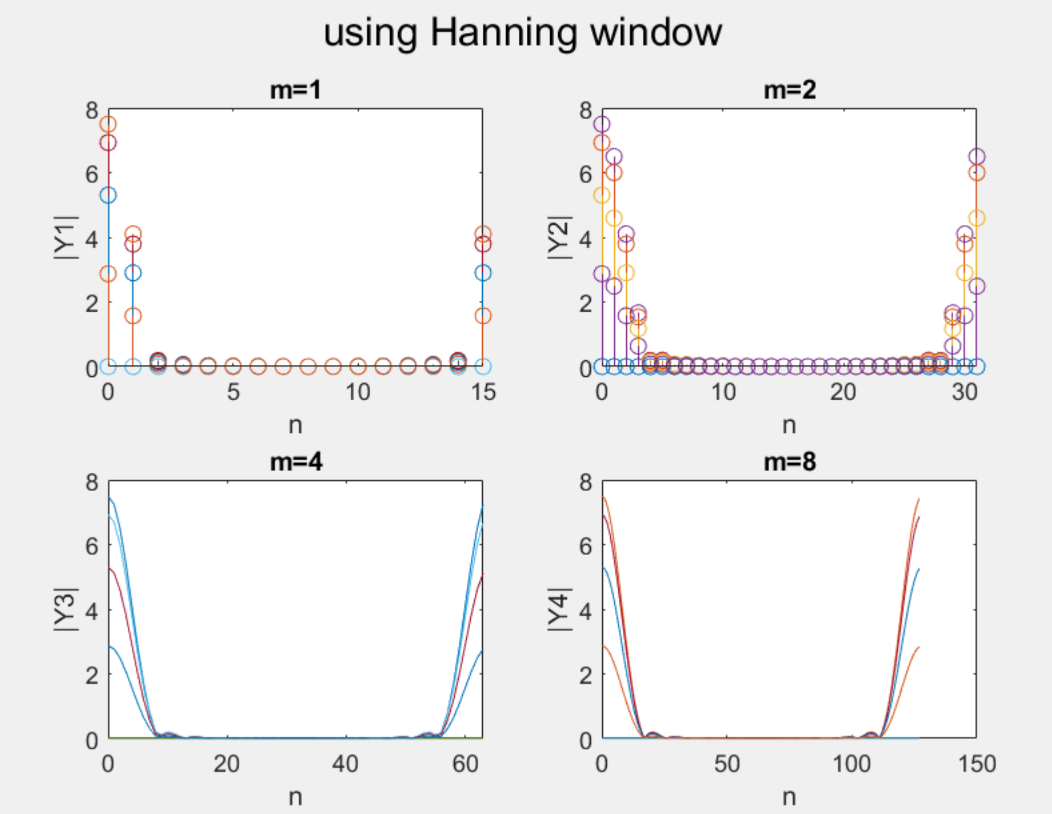


g)

code:

```
fo=12;
fs=64;
L=16;
n1=0:15;
n2=0:31;
n3=0:63;
n4=0:127;
pn1=cos((2*pi*fo*n1)/fs);
pn2=cos((2*pi*fo*n2)/fs);
pn3=cos((2*pi*fo*n3)/fs);
pn4=cos((2*pi*fo*n4)/fs);
wn1=hann(L);
wn2=hann(L);
wn3=hann(L);
wn4=hann(L);
xn1=pn1.*wn1;
xn2=pn2.*wn2;
xn3=pn3.*wn3;
xn4=pn4.*wn4;

y1=fft(xn1, 16);
y2=fft(xn2, 32);
y3=fft(xn3, 64);
y4=fft(xn4, 128);

figure;

subplot(2, 2, 1)
stem(n1, abs(y1));
title("m=1");
xlabel('n');
ylabel('|Y1|');

subplot(2, 2, 2)
stem(n2, abs(y2));
title("m=2");
xlabel('n');
ylabel('|Y2|');

subplot(2, 2, 3)
plot(n3, abs(y3));
title("m=4");
xlabel('n');
ylabel('|Y3|');

subplot(2, 2, 4)
plot(n4, abs(y4));
title("m=8");
xlabel('n');
ylabel('|Y4|');

sgtitle("using Hanning window")
```

plot:



Observation:

The Hanning window has a different spectral response compared to the rectangular window , affecting the mainlobe width and sidelobe levels .

Mainlobe width:

When we are using the Hanning window , the main lobe of the DFT magnitude spectrum will be narrower compared to the main lobe width observed in the part d with a rectangular window.

* The Hanning window results in better frequency localization , and the main lobe is more concentrated around the true frequency fo

Spectral Leakage:

*spectral leakage is reduced when a Hanning window is applied . The sidelobes in the DFT magnitude spectrum are lower in magnitude and spread over a narrower range of frequencies.

*compared to the rectangular window in part(d) , the Hanning window minimizes the spreading of energy from the mainlobe to adjacent frequency bins.

=> The Hanning window improves the spectral analysis by providing the better frequency resolution and reduced spectral leakage .

=> The main-lobe width becomes narrower , which may limit the ability to distinguish closely spaced frequency components.

h)

part d:

code:

```
f0 = 12;
fs = 64;
L = 16;
m_values = [1, 2, 4, 8];
f0_estimates = zeros(1, length(m_values));
n = 0:15;

for i = 1:length(m_values)
    m = m_values(i);
    N = m * L;
    pn = cos((2 * pi * f0 *n)/fs );
    wn=(0<=n & n<=15);
    xn=pn.*wn;
    X = fft(xn, N);
    f = (0:N-1) * fs / N;
    [~, index] = max(abs(X));
    f0_estimates(i) = f(index);
end

f0_estimates
```

Output:

```
    12      12      12      12
```

*In this case , the choice of N does not significantly affect the fo because there is minimal spectral leakage due to the rectangular window.

The peaks are appeared to be around f0=12Hz for all N values.

Part-f:

Code:

```
f0 = 11;
fs = 64;
L = 16;
m_values = [1, 2, 4, 8];
f0_estimates = zeros(1, length(m_values));
n = 0:15;

for i = 1:length(m_values)
    m = m_values(i);
    N = m * L;
    pn = cos((2 * pi * f0 *n)/fs );
    wn=(0<=n & n<=15);
    xn=pn.*wn;
    X = fft(xn, N);
    f = (0:N-1) * fs / N;
    [~, index] = max(abs(X));
    f0_estimates(i) = f(index);
end

f0_estimates
```

Output:

```
    12    10    11    11
```

Here the choice of N does not affect fo significantly due to minimal spectral leakage.

So the peaks appeared are around 11Hz.

Part-g:

```
f0 = 12;
fs = 64;
L = 16;
m_values = [1, 2, 4, 8];
f0_estimates = cell(1, length(m_values));

n = 0:15;

for i = 1:length(m_values)
    m = m_values(i);
    N = m * L;
    pn = cos((2 * pi * f0 * n) / fs);
    wn = hann(L);
    xn = pn .* wn;
    X = fft(xn, N);
    f = (0:N-1) * fs / N;
    [~, index] = max(abs(X));
    f0_estimates{i} = f(index);
end
for i = 1:length(m_values)
    disp(['m = ' num2str(m_values(i)) ', Estimated f0 = ' num2str(f0_estimates{i}) ' Hz']
end
```

The choice of N does not affect fo significantly due to reduced spectral leakage with the Hanning window.

Observation:

In parts d,f,g of this case are not affected fo significantly by the N because the spectral leakage is minimal or reduced by using the Hanning window , resultind in more accurate frequency estimation.

1)

i)

here I am loading the audio file numbered ‘1’ because the modulo of my roll no is ‘1’(ie modulo(10) of 2022102021 is ‘1’).so by using the below code

```
audioFileName = '1.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

I am getting the first three strongest frequencies are

```
Three Strongest Frequencies (in Hz):
   10.0000  990.0000   10.0125
```

FOR OTHER PARTS:

For file named ‘0’

Code:

```
audioFileName = '0.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
    5.0000   995.0000     5.0125
```

*For file named ‘2’

Code:

```
audioFileName = '2.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   15.0000   985.0000    15.0125
```

*For file named ‘3’

Code:

```
audioFileName = '3.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   20.0000   980.0000    20.0125
```

*For file named ‘4’

Code:

```
audioFileName = '4.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   25.0000  975.0000   25.0125
```

*For file named ‘5’

Code:

```
audioFileName = '5.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   30.0000  970.0000   30.0125
```

*For file named ‘6’

Code:

```
audioFileName = '6.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   35.0000  965.0000   34.9875
```

*For file named '7'

Code:

```
audioFileName = '7.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   40.0000  960.0000   40.0125
```

*For file named '8'

Code:

```
audioFileName = '8.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

```
Three Strongest Frequencies (in Hz):
   45.0000  955.0000   45.0125
```

*For file named '9'

Code:

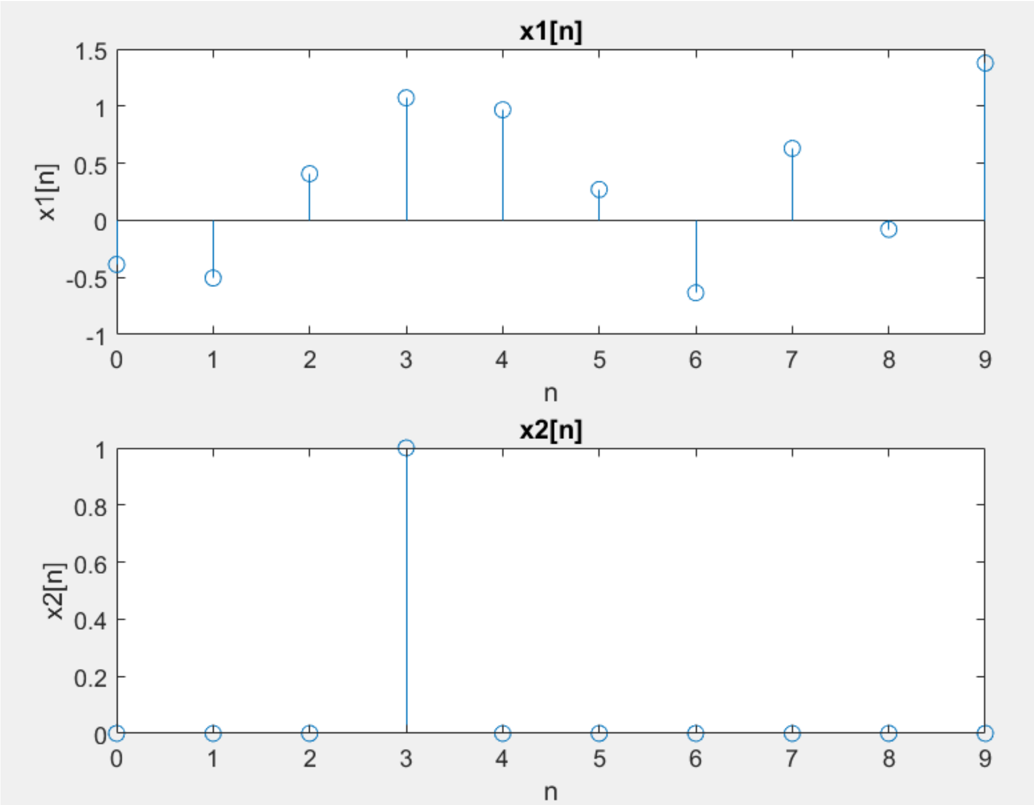
```
audioFileName = '9.wav';
[y, fs] = audioread(audioFileName);
L = length(y);
N = 8 * L;
Y = fft(y, N);
f = (0:N-1) * fs / N;
[sortedValues, sortedIndices] = sort(abs(Y), 'descend');
strongestFrequencies = f(sortedIndices(1:3));
disp('Three Strongest Frequencies (in Hz):');
disp(strongestFrequencies);
```

Output:

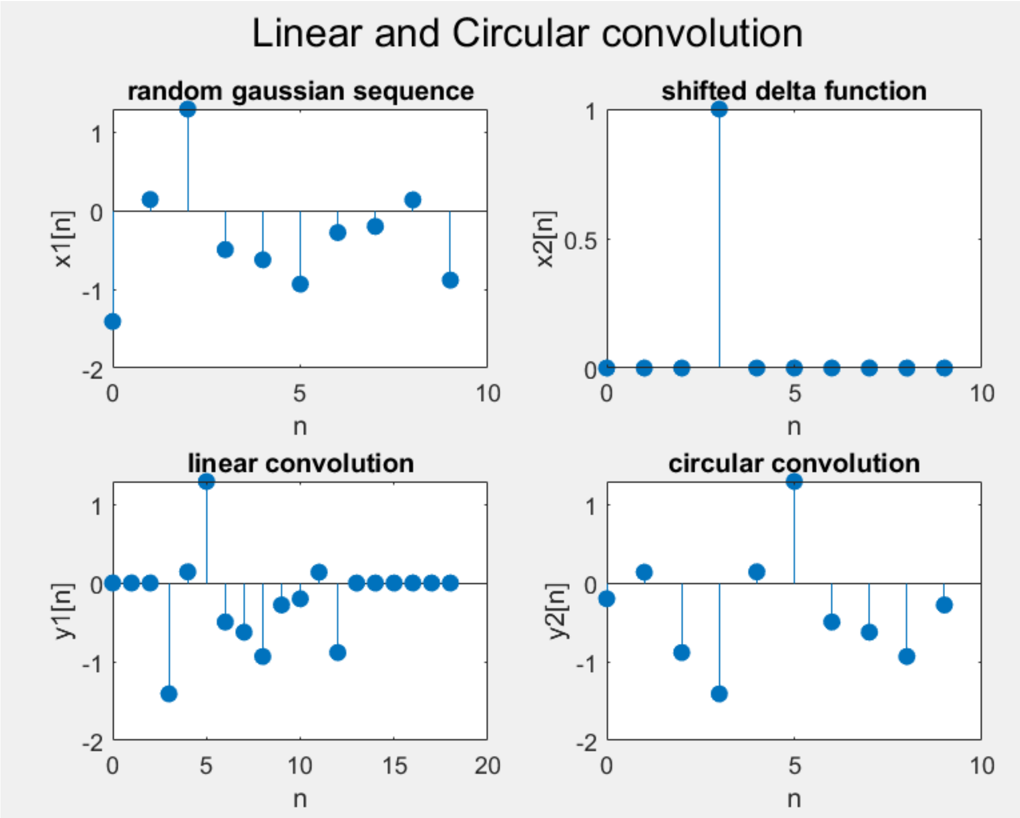
```
Three Strongest Frequencies (in Hz):
   50.0000  950.0000   49.9875
```

2)

a)

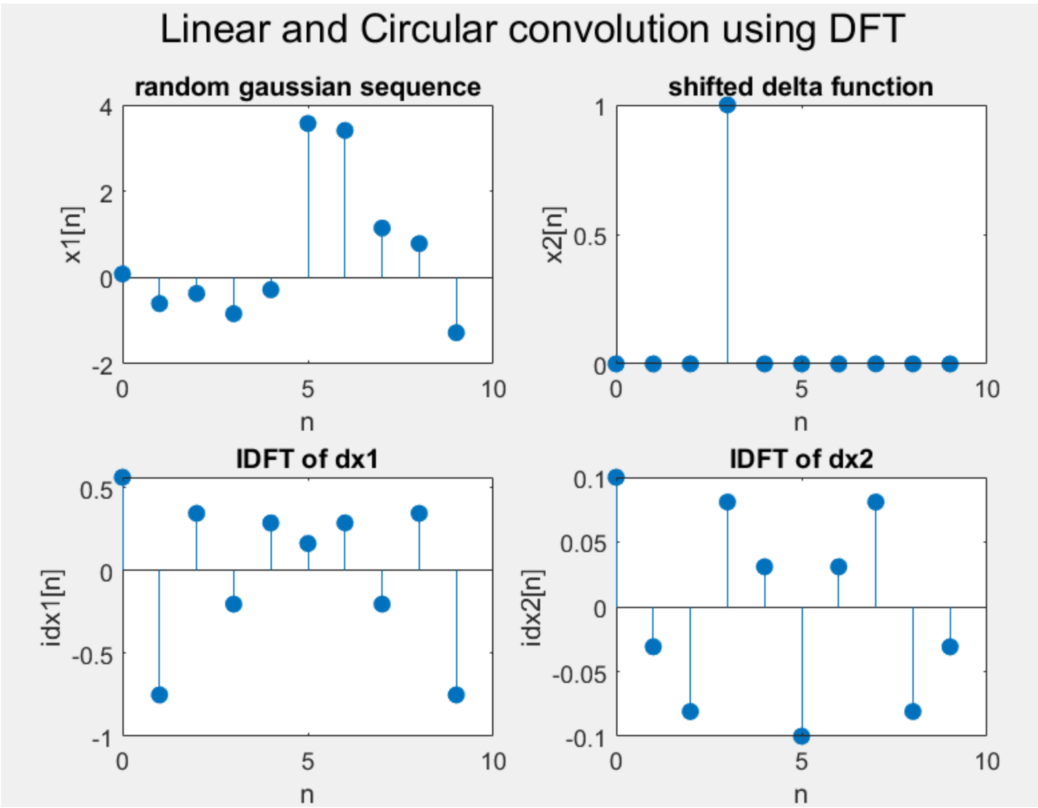

x1[n]
1.5
1
0.5
0
-0.5
-1
x1[n]
0 1 2 3 4 5 6 7 8 9
n
x2[n]
1
0.8
0.6
0.4
0.2
0
x2[n]
0 1 2 3 4 5 6 7 8 9
n

b)



Linear and Circular convolution
random gaussian sequence
shifted delta function
linear convolution
circular convolution
x1[n]
x2[n]
y1[n]
y2[n]
n

c)



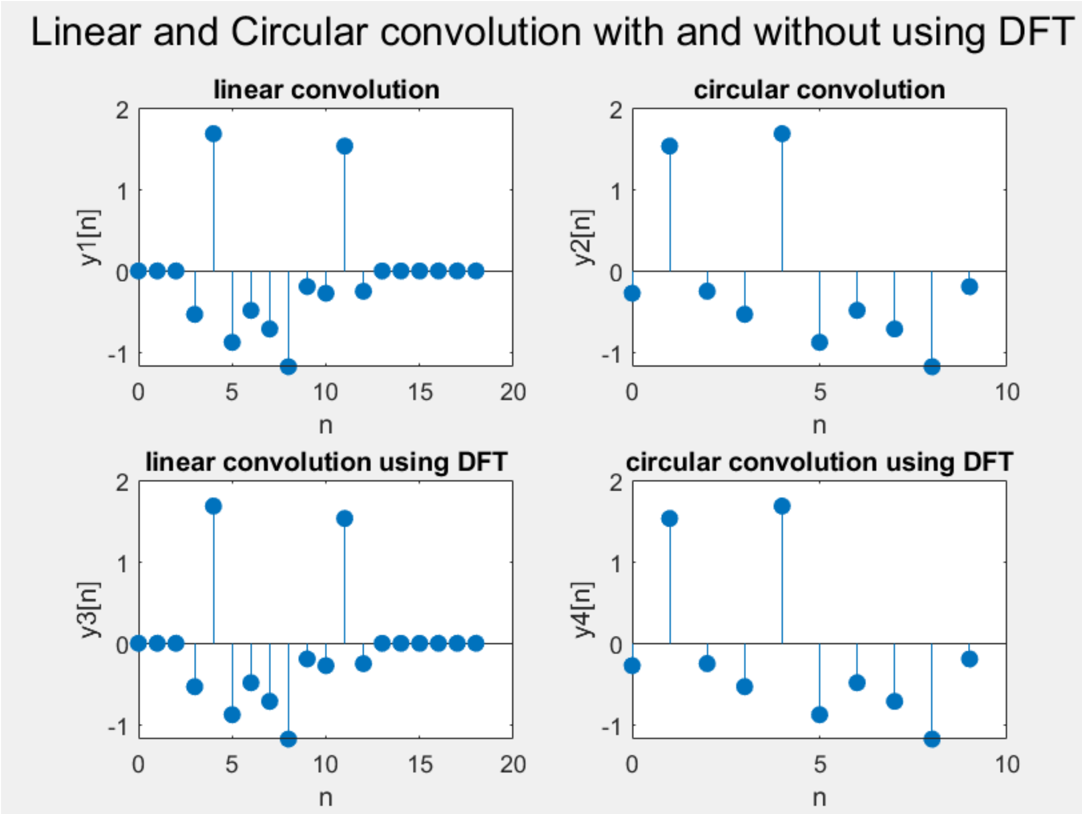
Linear and Circular convolution using DFT
random gaussian sequence
shifted delta function
IDFT of dx1
IDFT of dx2
x1[n]
x2[n]
idx1[n]
idx2[n]
n

d)

CODE:

```
n=0:9;
x1=randn(1,10);
% x2=[0,0,0,1,0,0,0,0,0,0];
x2=(n==3);

dx1=fft(x1);
dx2 =fft(x2);
idx1=ifft(dx1);
idx2=ifft(dx2);
y1=conv(x1,x2);
y2=cconv(x1,x2);
y3=conv(idx1,idx2);
y4=cconv(idx1,idx2);
c_n = 0:length(y2)-1;

figure

subplot(2,2,1);
stem(0:18, y1, "filled");
title('linear convolution');
xlabel('n');
ylabel('y1[n]');

subplot(2,2,2);
stem(c_n, y2, "filled");
title('circular convolution');
xlabel('n');
ylabel('y2[n]');


subplot(2,2,3);
stem(0:18, y3, "filled");
title('linear convolution using DFT');
xlabel('n');
ylabel('y3[n]');

subplot(2,2,4);
stem(c_n, y4, "filled");
title('circular convolution using DFT');
xlabel('n');
ylabel('y4[n]');

sgtitle('Linear and Circular convolution with and without using DFT');
```

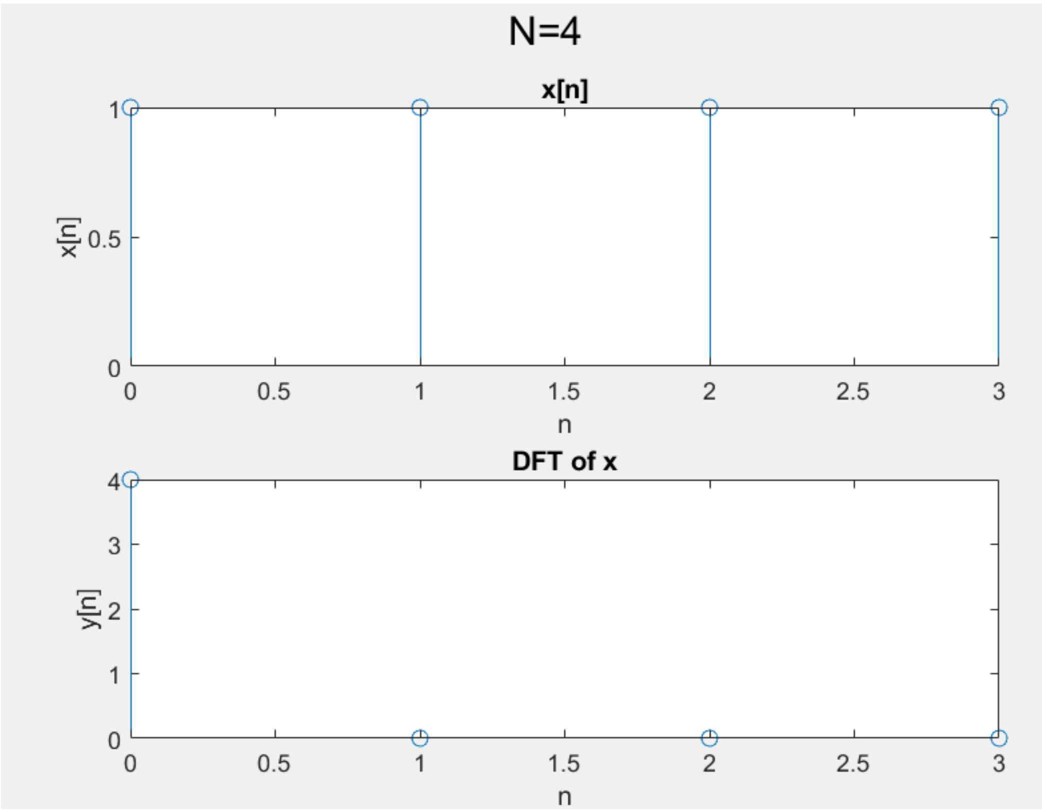
PLOT:

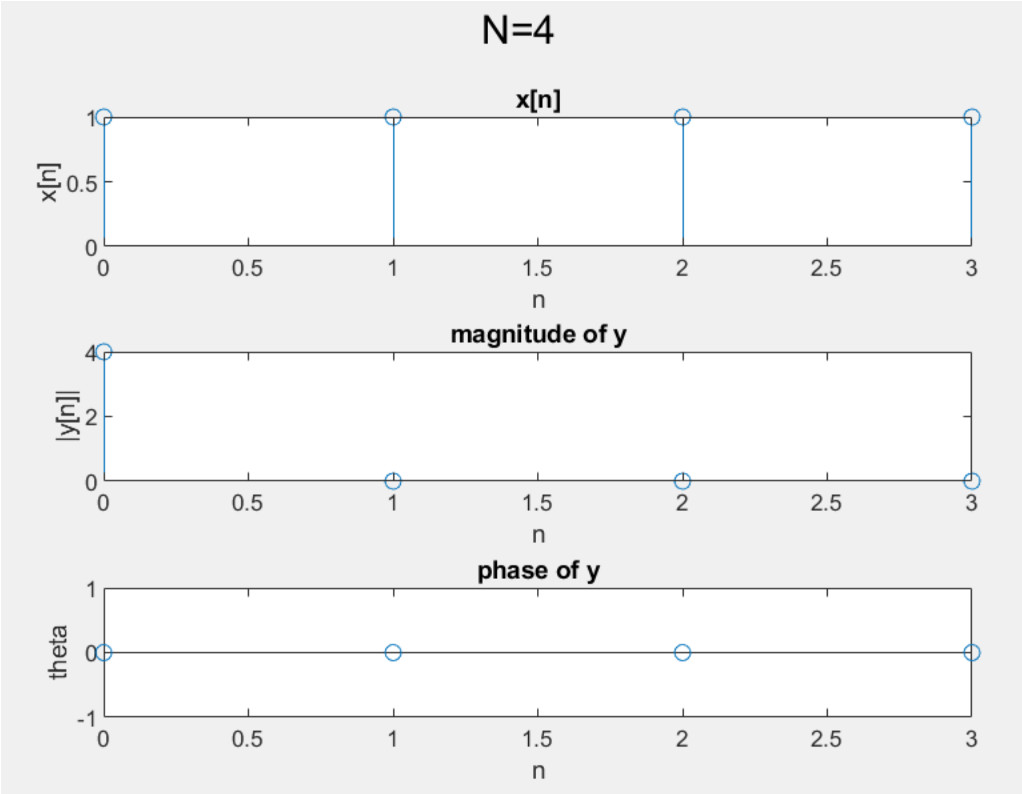

Linear and Circular convolution with and without using DFT
linear convolution
circular convolution
linear convolution using DFT
circular convolution using DFT
y1[n]
y2[n]
y3[n]
y4[n]
n

3)

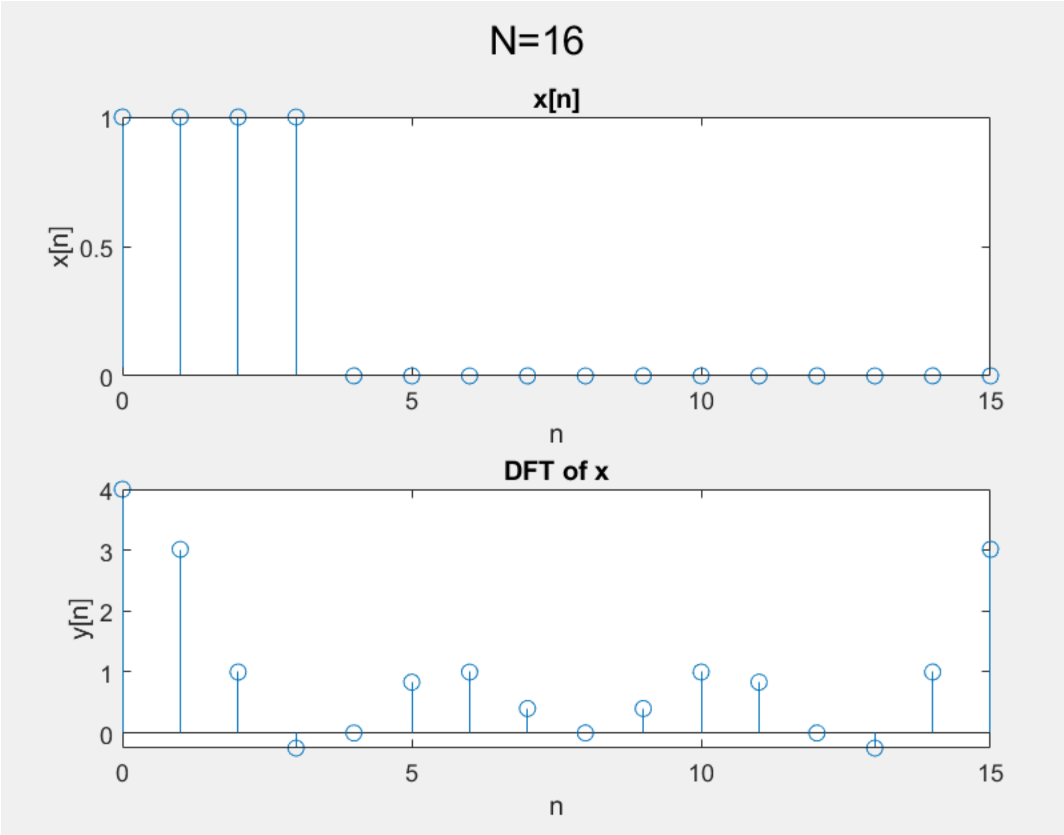
a)

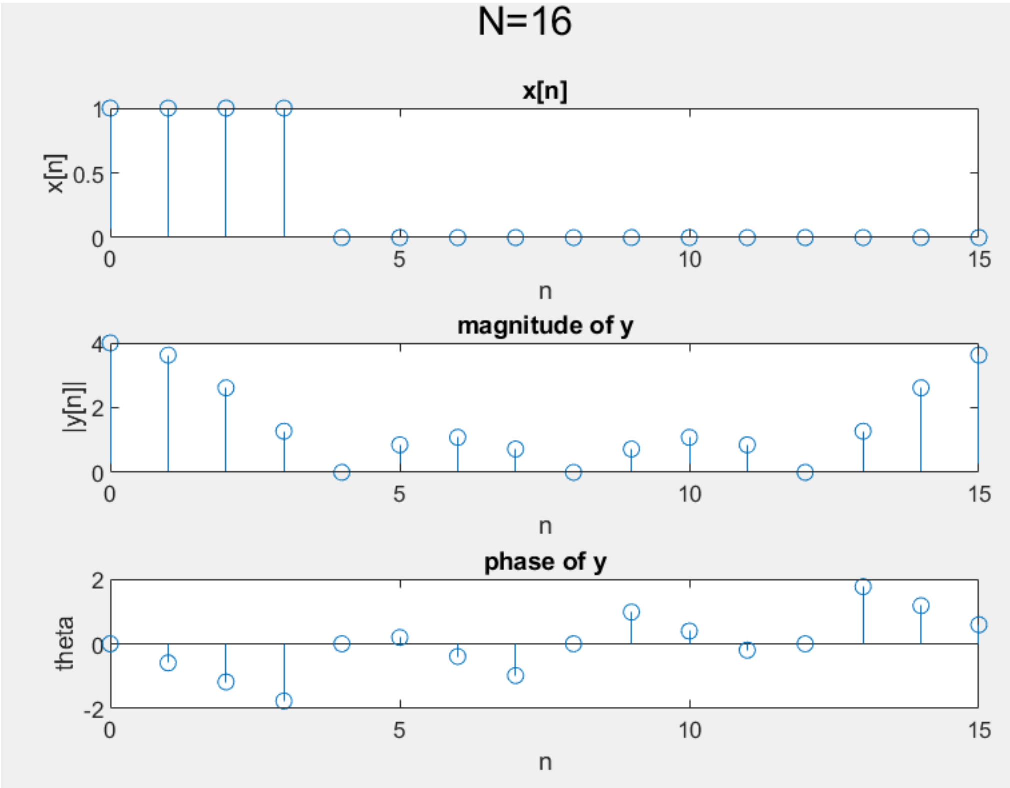
when N=4


N=4
x[n]
DFT of x
y[n]
n

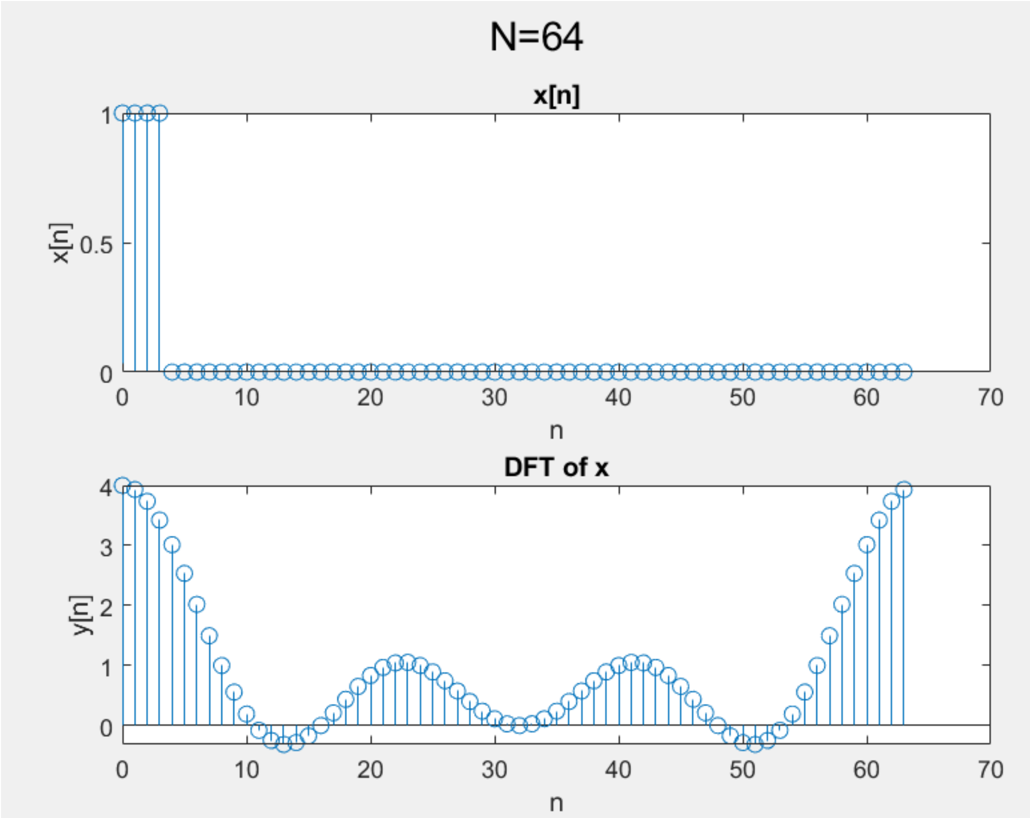

N=4
x[n]
x[n]
1
0.5
0
0
0.5
1
1.5
2
2.5
3
n
magnitude of y
|y[n]|
4
2
0
n
phase of y
theta
1
0
-1
n

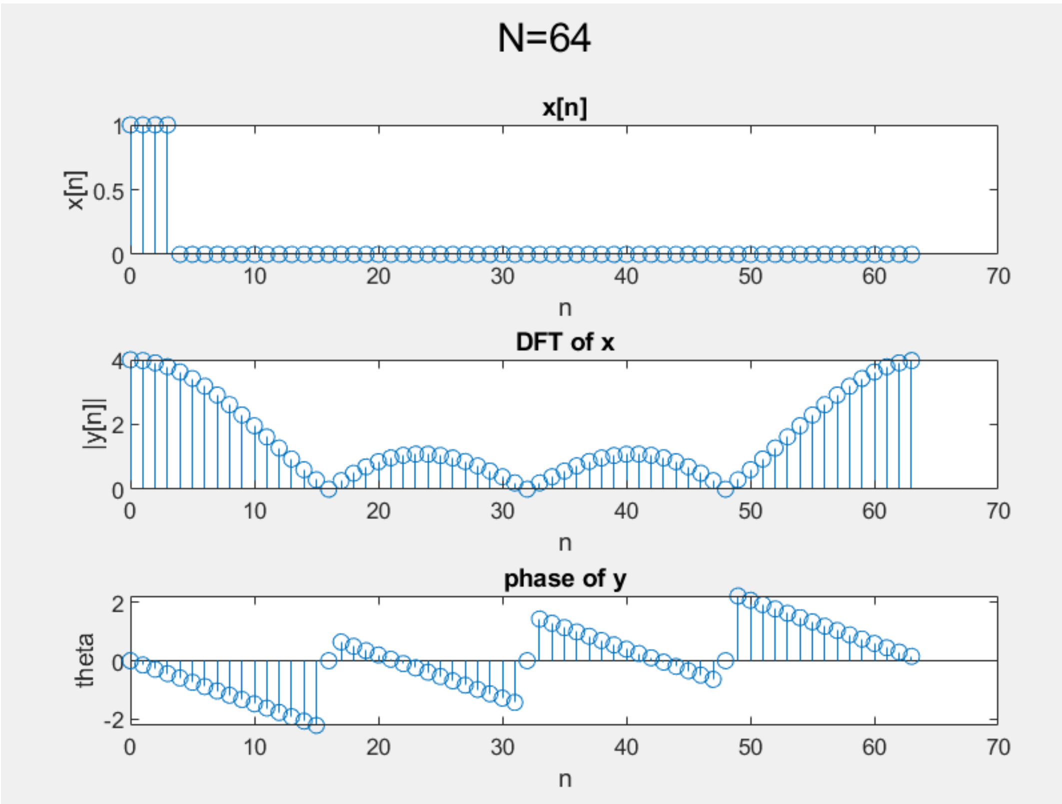
When N=16


N=16
x[n]
x[n]
n
DFT of x
y[n]
n


N=16
x[n]
x[n]
n
magnitude of y
|y[n]|
n
phase of y
theta
n

When N=64


N=64
x[n]
x[n]
n
DFT of x
y[n]
n

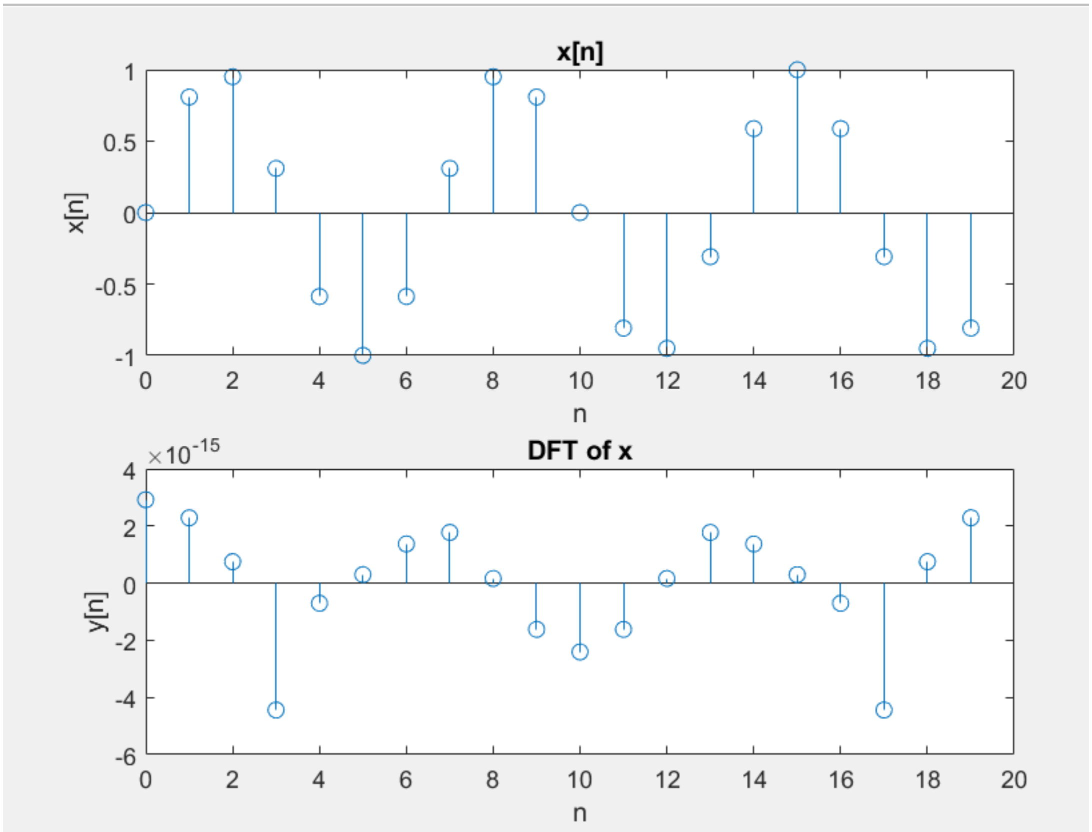

N=64
x[n]
x[n]
n
DFT of x
|y[n]|
n
phase of y
theta
n

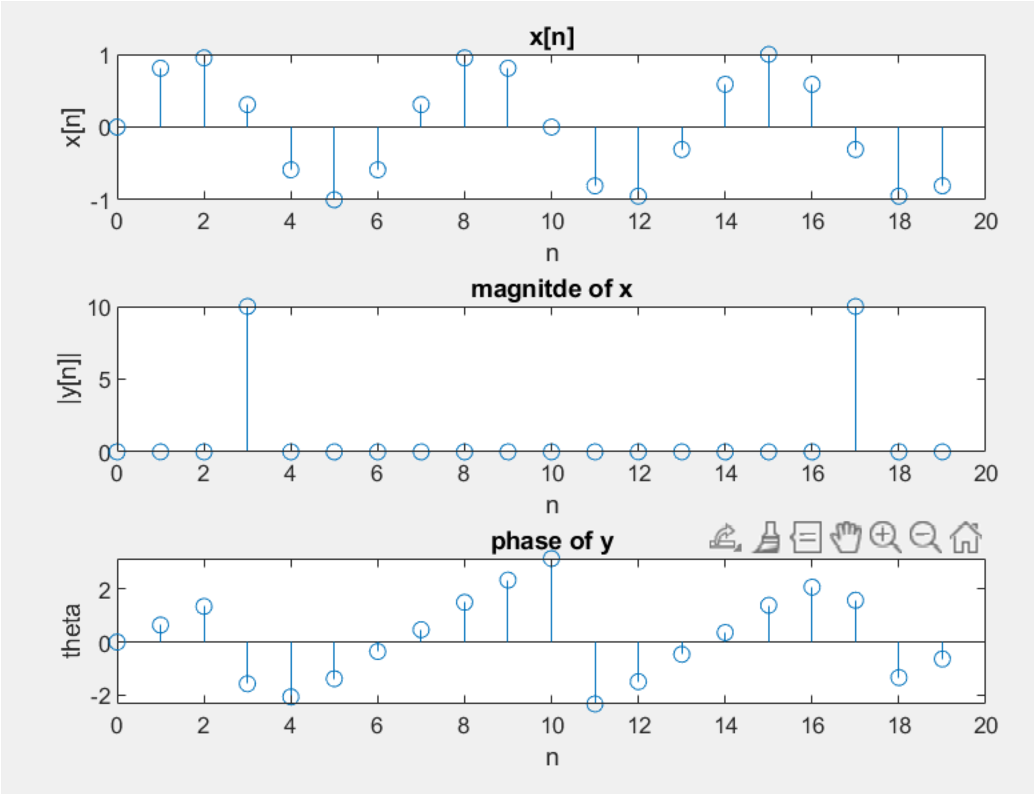
b)

CODE:

```
N=20;
n=0:N-1;
wo=(3*pi)/10;
x=sin(wo*n);
y=fft(x,N);
figure
subplot(3,1,1)
stem(n,x);
title('x[n]')
xlabel('n');
ylabel('x[n]');
subplot(3,1,2)
stem(n,abs(y));
title('absolute of DFT of x')
xlabel('n');
ylabel('|y[n]|');
subplot(3,1,3)
stem(n,angle(y));
title('phase of y')
xlabel('n');
ylabel('theta');
```

Plots:



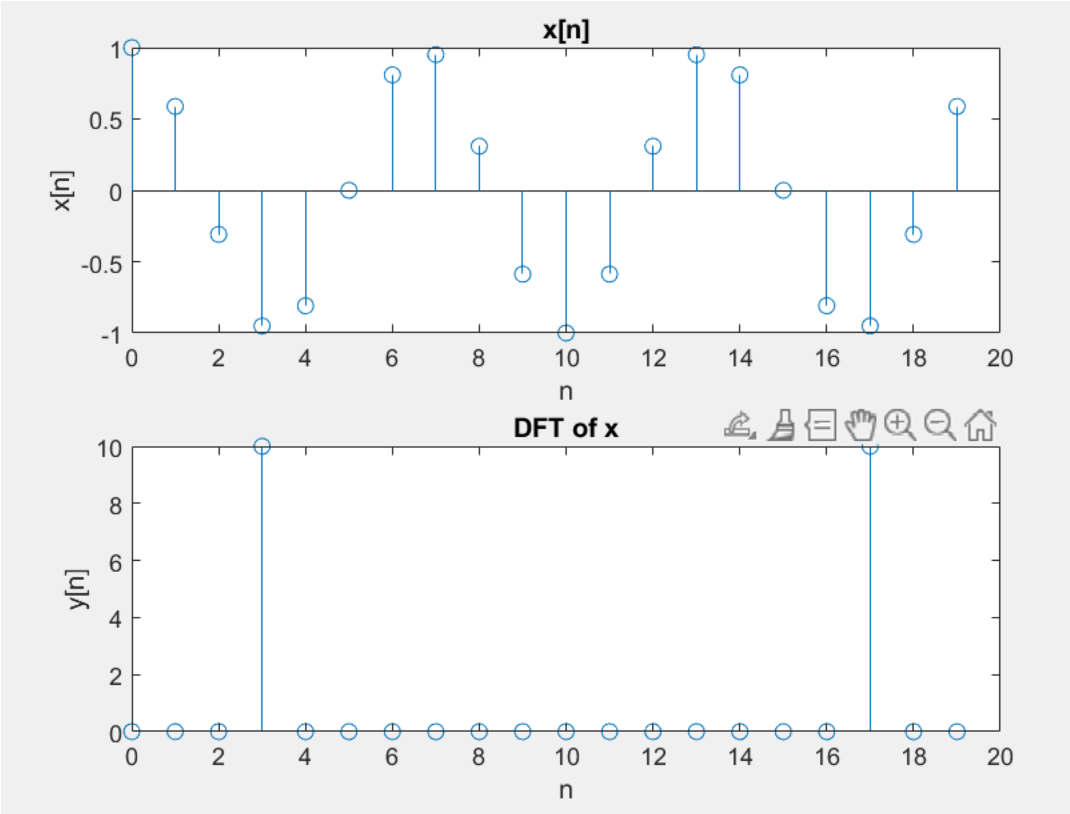


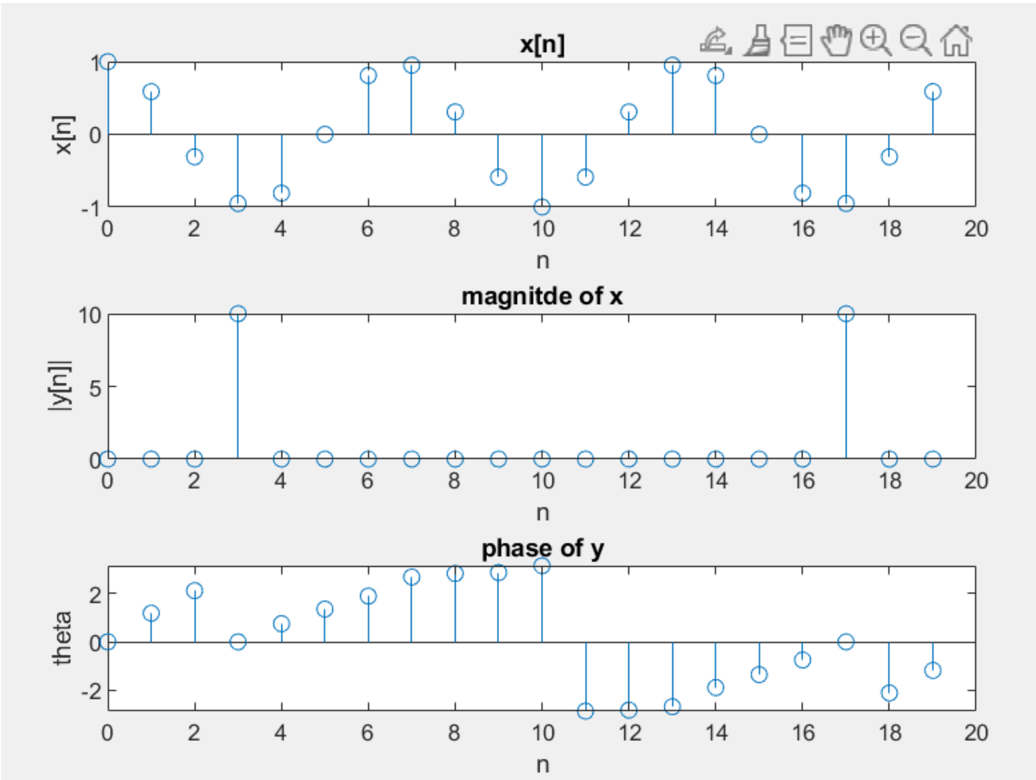
c)

CODE:

```
N=20;
n=0:N-1;
wo=(3*pi)/10;
x=cos(wo*n);
y=fft(x,N);
figure
subplot(3,1,1)
stem(n,x);
title('x[n]')
xlabel('n');
ylabel('x[n]');
subplot(3,1,2)
stem(n,abs(y));
title('absolute of DFT of x')
xlabel('n');
ylabel('|y[n]|');
subplot(3,1,3)
stem(n,angle(y));
title('phase of y')
xlabel('n');
ylabel('theta');
```

Plots:



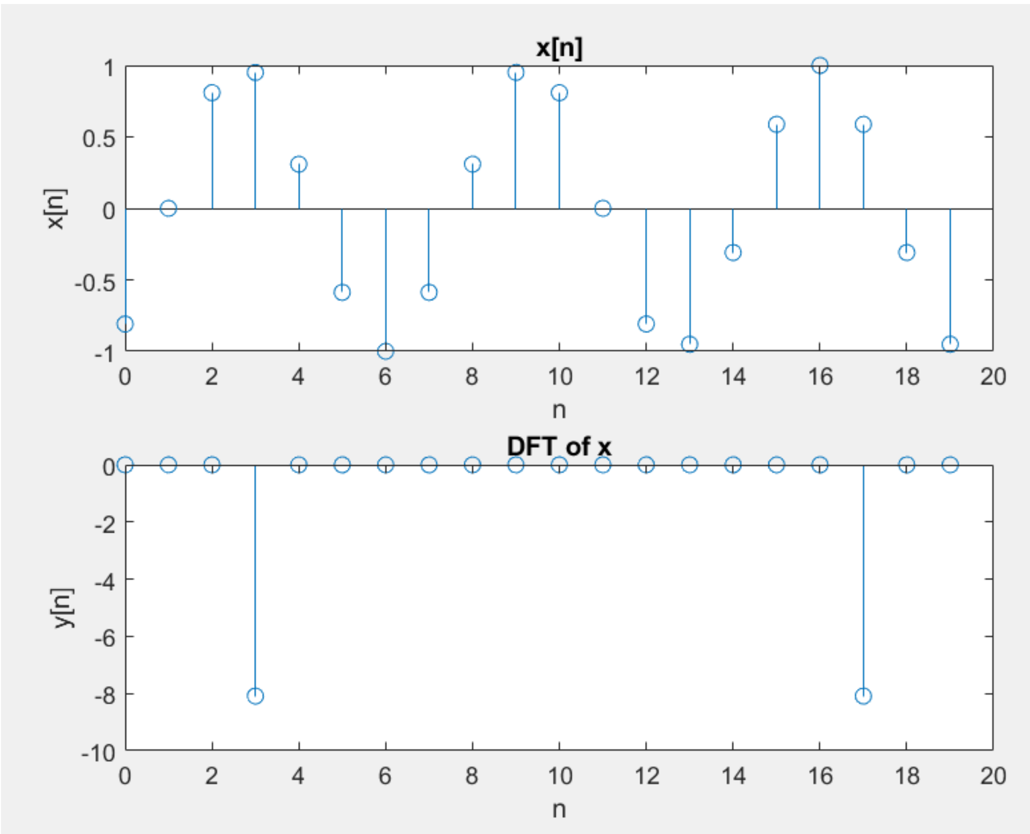


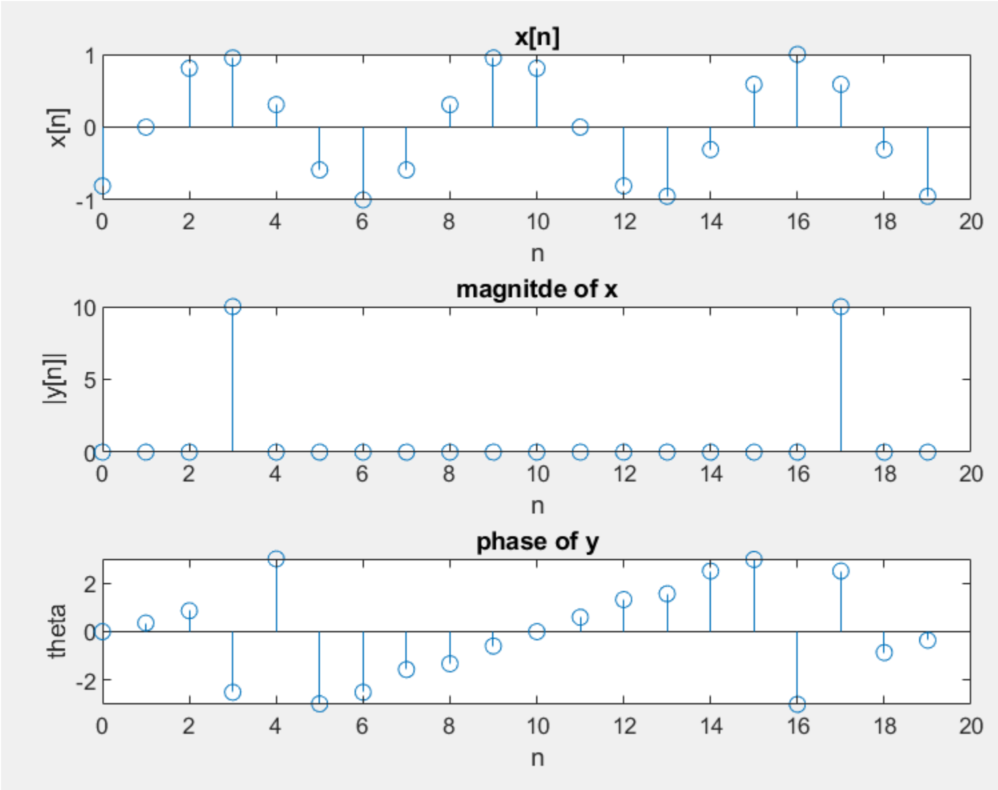
d)

CODE:

```
N=20;
n=0:N-1;
wo=(3*pi)/10;
x=sin(wo*(n-1));
y=fft(x,N);
figure
subplot(3,1,1)
stem(n,x);
title('x[n]')
xlabel('n');
ylabel('x[n]');
subplot(3,1,2)
stem(n,abs(y));
title('absolute of DFT of x')
xlabel('n');
ylabel('|y[n]|');
subplot(3,1,3)
stem(n,angle(y));
title('phase of y')
xlabel('n');
ylabel('theta');
```

Plots:



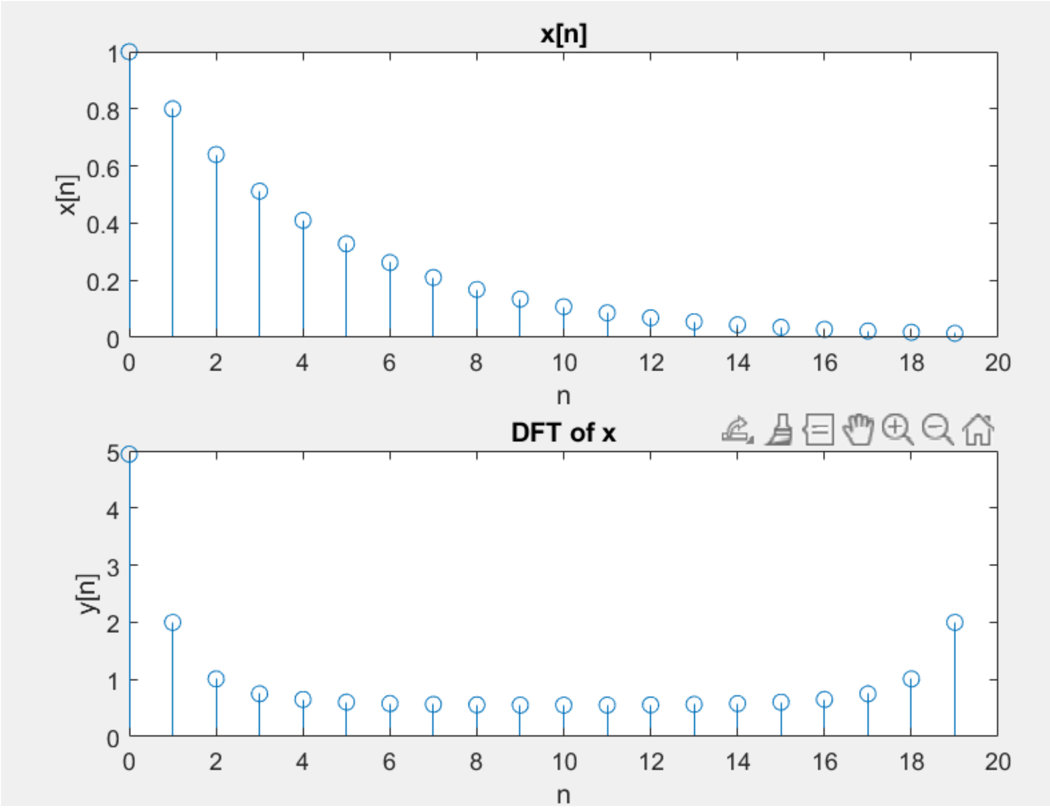


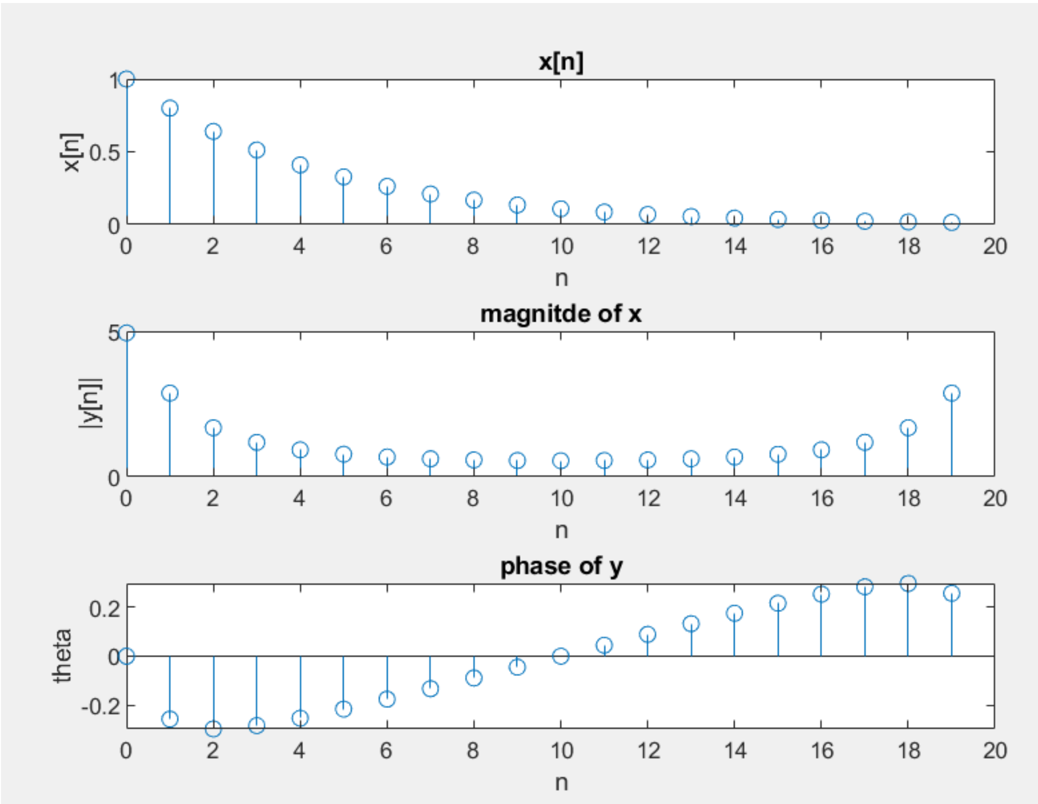
e)

CODES:

```
N=20;
n=0:N-1;
wo=(3*pi)/10;
x=(0.8).^n;
y=fft(x,N);
figure
subplot(3,1,1)
stem(n,x);
title('x[n]')
xlabel('n');
ylabel('x[n]');
subplot(3,1,2)
stem(n,abs(y));
title('absolute of DFT of x')
xlabel('n');
ylabel('|y[n]|');
subplot(3,1,3)
stem(n,angle(y));
title('phase of y')
xlabel('n');
ylabel('theta');
```

Plots:



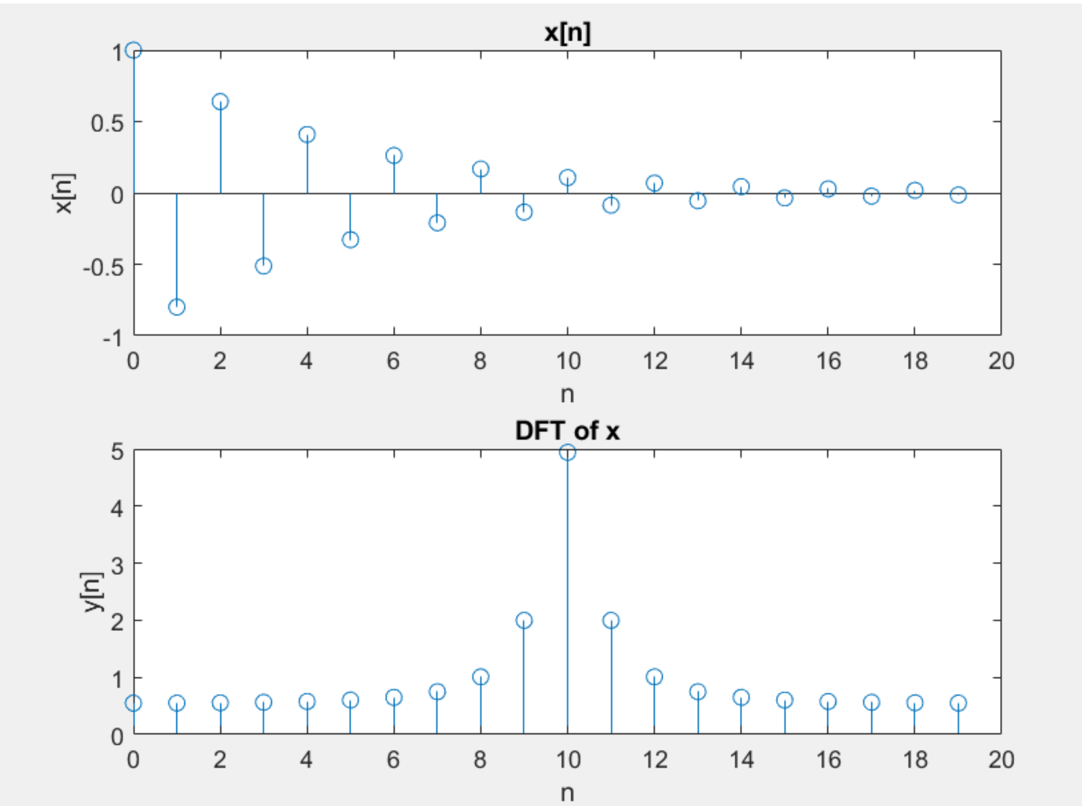


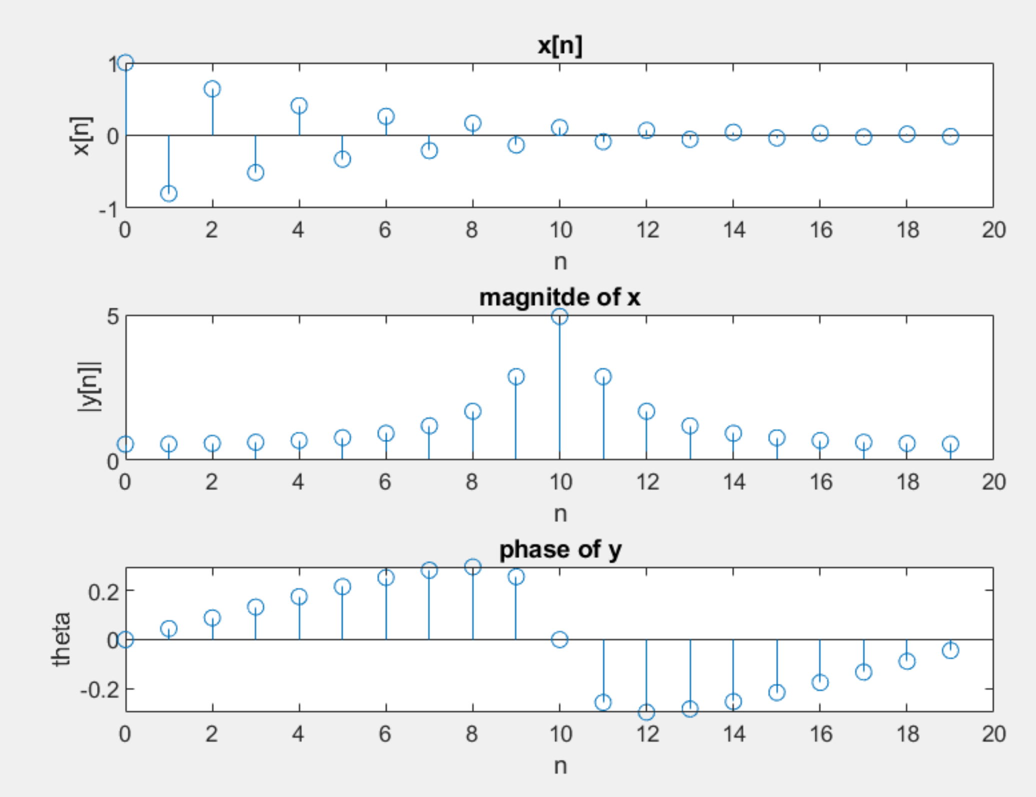
f)

CODE:

```
N=20;
n=0:N-1;
wo=(3*pi)/10;
x=(-1*0.8).^n;
y=fft(x,N);
figure
subplot(3,1,1)
stem(n,x);
title('x[n]')
xlabel('n');
ylabel('x[n]');
subplot(3,1,2)
stem(n,abs(y));
title('absolute of DFT of x')
xlabel('n');
ylabel('|y[n]|');
subplot(3,1,3)
stem(n,angle(y));
title('phase of y')
xlabel('n');
ylabel('theta');
```

Plots:





We can find the low-frequency and high-frequency components that contribute most significantly to the signal by analysing the magnitude spectrum .

The peaks or significant values that are closer to the zero frequency are associated with low – frequency components , while those farther from the zero frequency are associated with high-frequency components.