PRACTICAL 4 : DFT and IDFT

U19EC046 - Harsh Suthar **Date: XX-XX-XXXX**

AIM

To find the discrete Fourier transform of given sequence.

THEORY

The discrete Fourier transform (DFT) is a method for converting a sequence of complex numbers sequence of complex numbers,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n/N},$$

$0 \leq k \leq N-1$ For

The are thought of as the values of a function, or signal, at equally spaced times The output complex number which encodes the amplitude and phase of a sinusoidal wave with frequency cycles per time un comes from Euler's formula:

$$e^{2\pi i k n/N} = \cos(2\pi k n/N) + i \sin(2\pi k n/N).$$

The effect of computing the is to fi coefficients of an approximation of the signal by a linear combination of such waves. Since each wave has an integer cycles per time units, the approximation will be periodic with period This approximation is given by the inverse transform

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{2\pi i k n/N}.$$

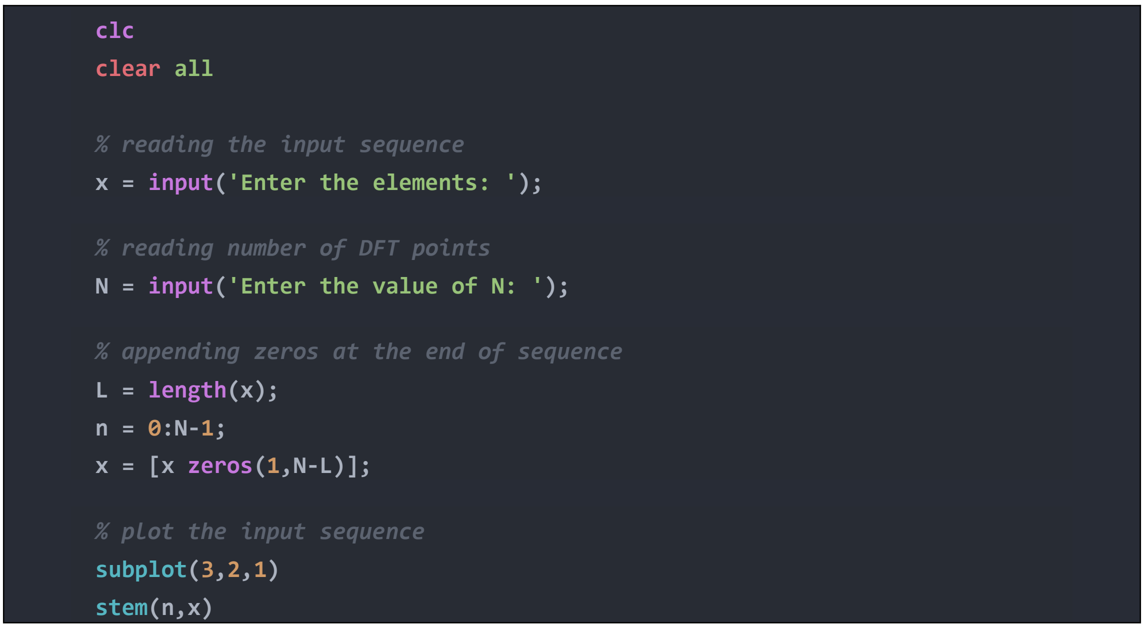
The DFT is useful in many applications, including the simple signal spectral analysis outlined above. Knowing how a s be expressed as a combination of waves allows for manipulation of that signal and comparisons of different signals

ALGORITHM

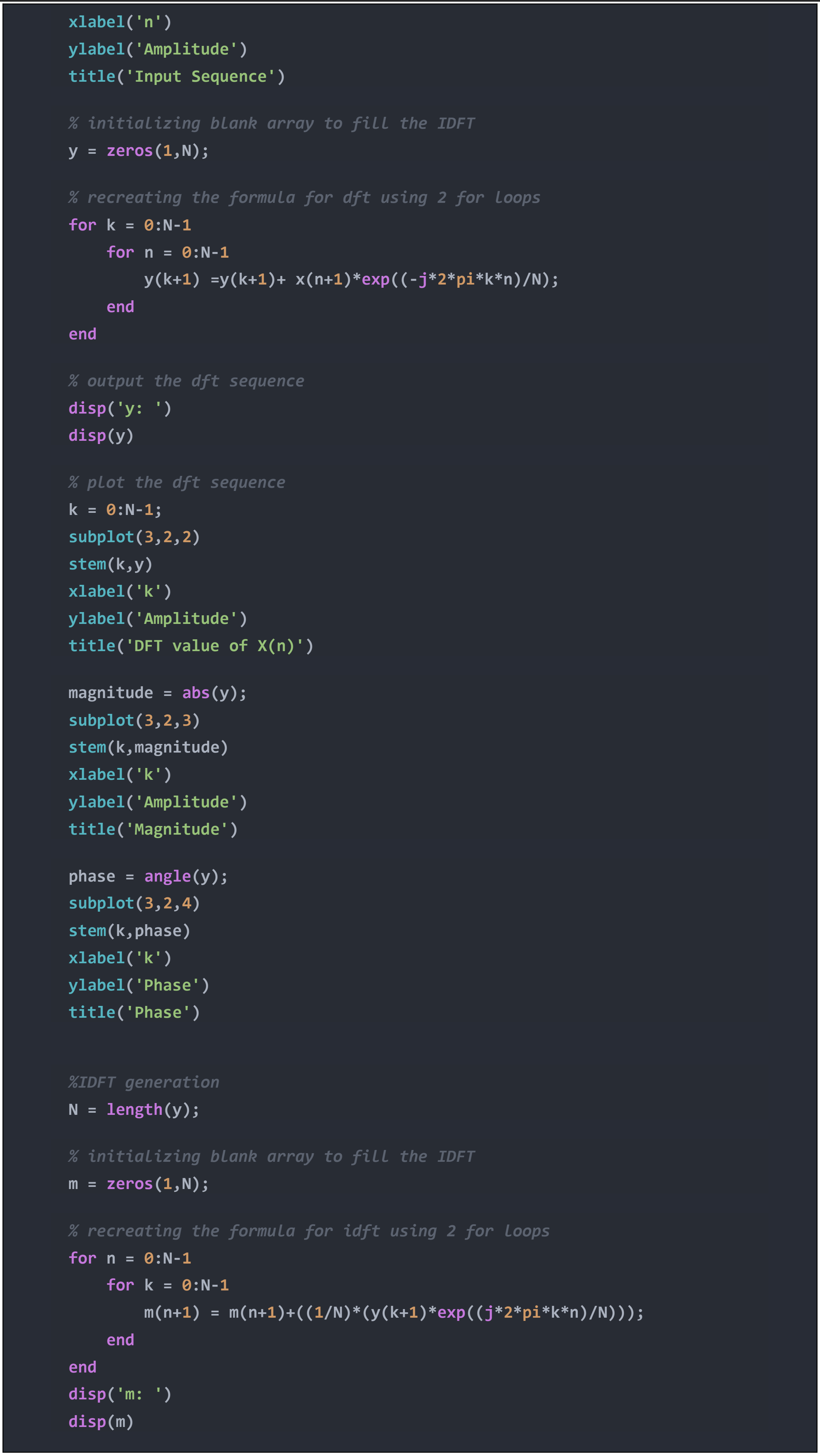
1. Read the input for sequence and number of points for dft from the user.
2. Initialize a blank array of size equal to number of dft points, to be filled with the idft sequence.
3. Create two for loops, the outer for loop will fill the idft sequence over the value of (1, N) in the array, the inner for loop will perform the summation part for a particular value of n, effectively recreating the above formula.
4. Finally, display and plot the generated idft sequence.

CODE

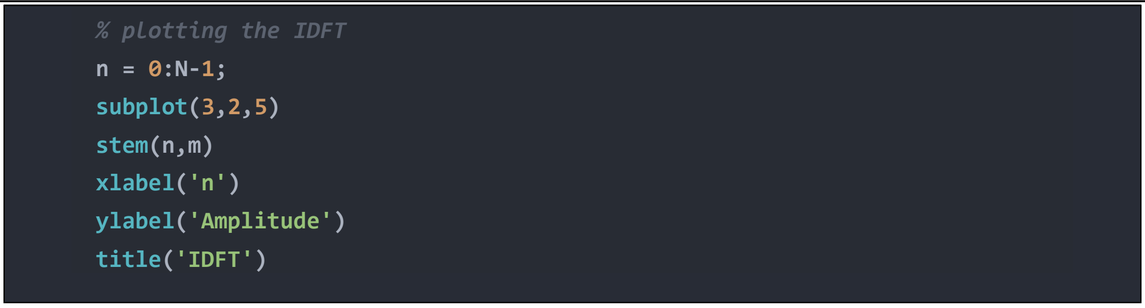
1. Without using in-built function



```
clc
clear all

% reading the input sequence
x = input('Enter the elements: ');

% reading number of DFT points
N = input('Enter the value of N: ');

% appending zeros at the end of sequence
L = length(x);
n = 0:N-1;
x = [x zeros(1,N-L)];

% plot the input sequence
subplot(3,2,1)
stem(n,x)
```

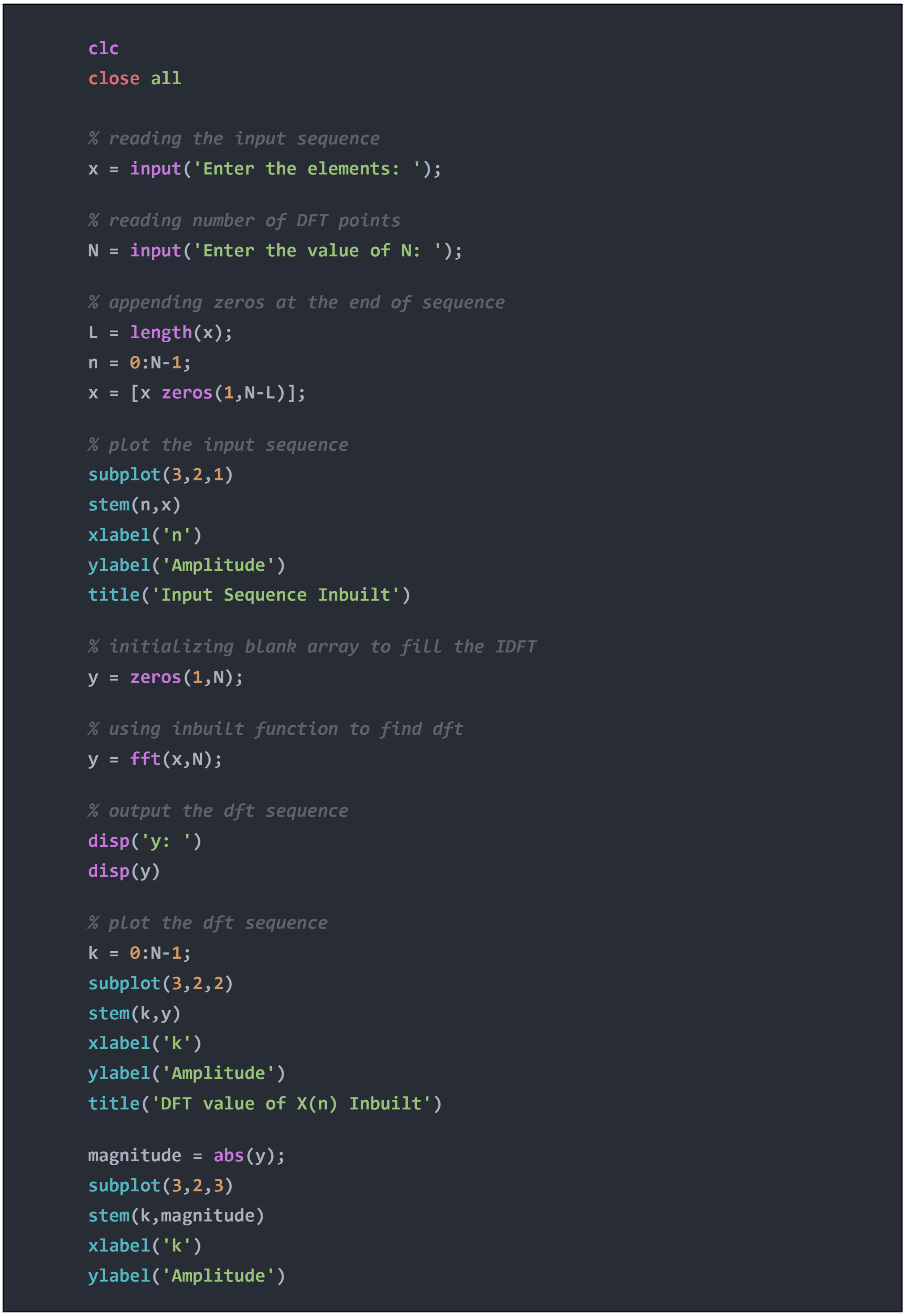


```
xlabel('n')
ylabel('Amplitude')
title('Input Sequence')

% initializing blank array to fill the IDFT
y = zeros(1,N);

% recreating the formula for dft using 2 for loops
for k = 0:N-1
    for n = 0:N-1
        y(k+1) =y(k+1)+ x(n+1)*exp((-j*2*pi*k*n)/N);
    end
end

% output the dft sequence
disp('y: ')
disp(y)

% plot the dft sequence
k = 0:N-1;
subplot(3,2,2)
stem(k,y)
xlabel('k')
ylabel('Amplitude')
title('DFT value of X(n)')

magnitude = abs(y);
subplot(3,2,3)
stem(k,magnitude)
xlabel('k')
ylabel('Amplitude')
title('Magnitude')

phase = angle(y);
subplot(3,2,4)
stem(k,phase)
xlabel('k')
ylabel('Phase')
title('Phase')


%IDFT generation
N = length(y);

% initializing blank array to fill the IDFT
m = zeros(1,N);

% recreating the formula for idft using 2 for loops
for n = 0:N-1
    for k = 0:N-1
        m(n+1) = m(n+1)+((1/N)*(y(k+1)*exp((j*2*pi*k*n)/N)));
    end
end
disp('m: ')
disp(m)
```

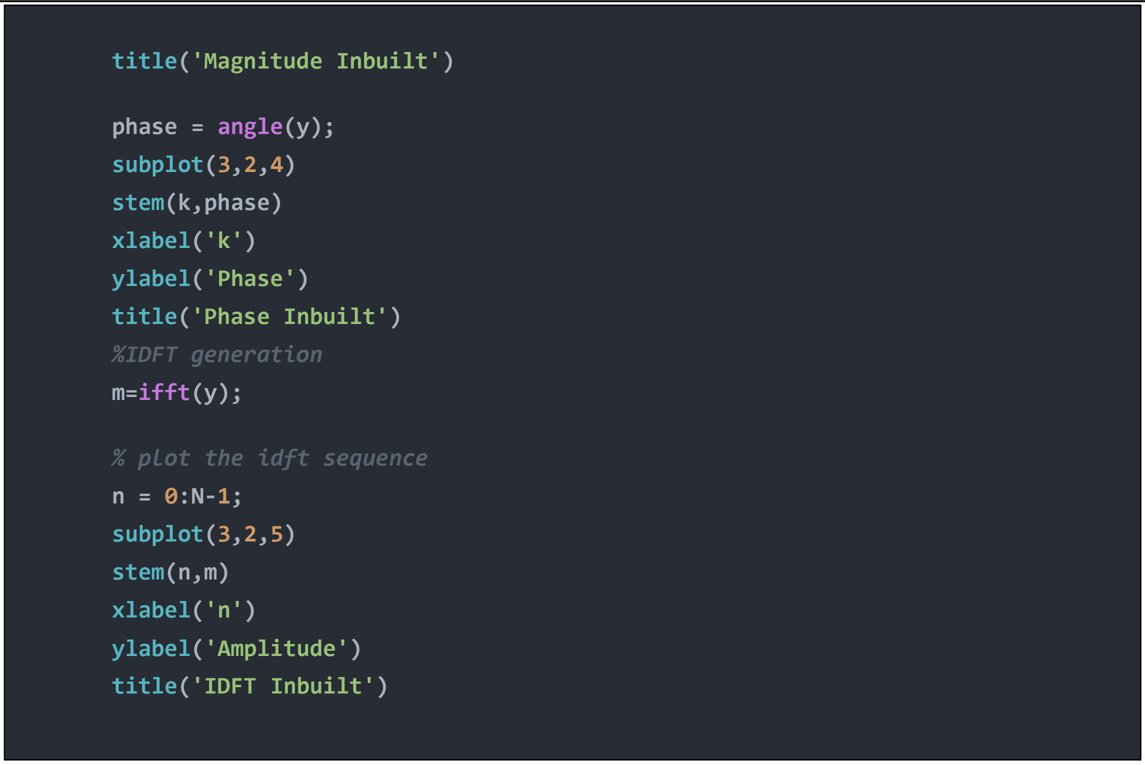


```
% plotting the IDFT
n = 0:N-1;
subplot(3,2,5)
stem(n,m)
xlabel('n')
ylabel('Amplitude')
title('IDFT')
```

2. Using in-built function



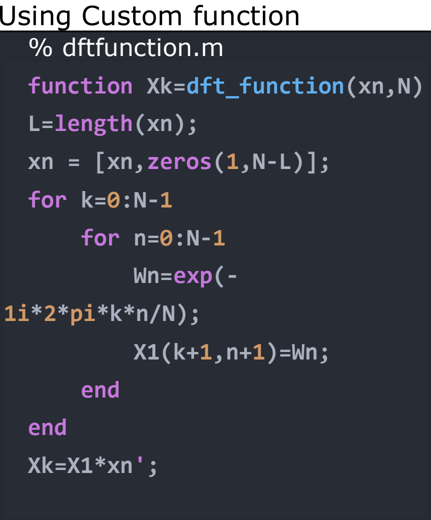
```
clc
close all

% reading the input sequence
x = input('Enter the elements: ');

% reading number of DFT points
N = input('Enter the value of N: ');

% appending zeros at the end of sequence
L = length(x);
n = 0:N-1;
x = [x zeros(1,N-L)];

% plot the input sequence
subplot(3,2,1)
stem(n,x)
xlabel('n')
ylabel('Amplitude')
title('Input Sequence Inbuilt')

% initializing blank array to fill the IDFT
y = zeros(1,N);

% using inbuilt function to find dft
y = fft(x,N);

% output the dft sequence
disp('y: ')
disp(y)

% plot the dft sequence
k = 0:N-1;
subplot(3,2,2)
stem(k,y)
xlabel('k')
ylabel('Amplitude')
title('DFT value of X(n) Inbuilt')

magnitude = abs(y);
subplot(3,2,3)
stem(k,magnitude)
xlabel('k')
ylabel('Amplitude')
```



```
title('Magnitude Inbuilt')

phase = angle(y);
subplot(3,2,4)
stem(k,phase)
xlabel('k')
ylabel('Phase')
title('Phase Inbuilt')
%IDFT generation
m=ifft(y);

% plot the idft sequence
n = 0:N-1;
subplot(3,2,5)
stem(n,m)
xlabel('n')
ylabel('Amplitude')
title('IDFT Inbuilt')
```

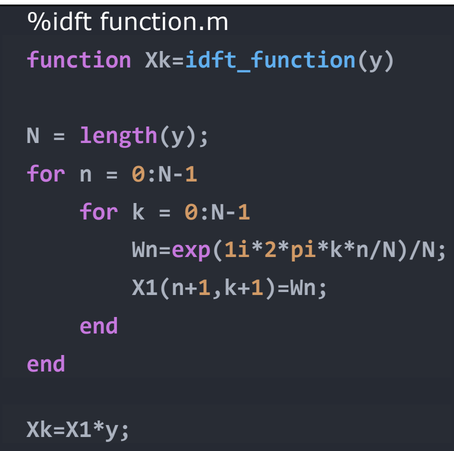
3. Using Custom function

% dftfunction.m

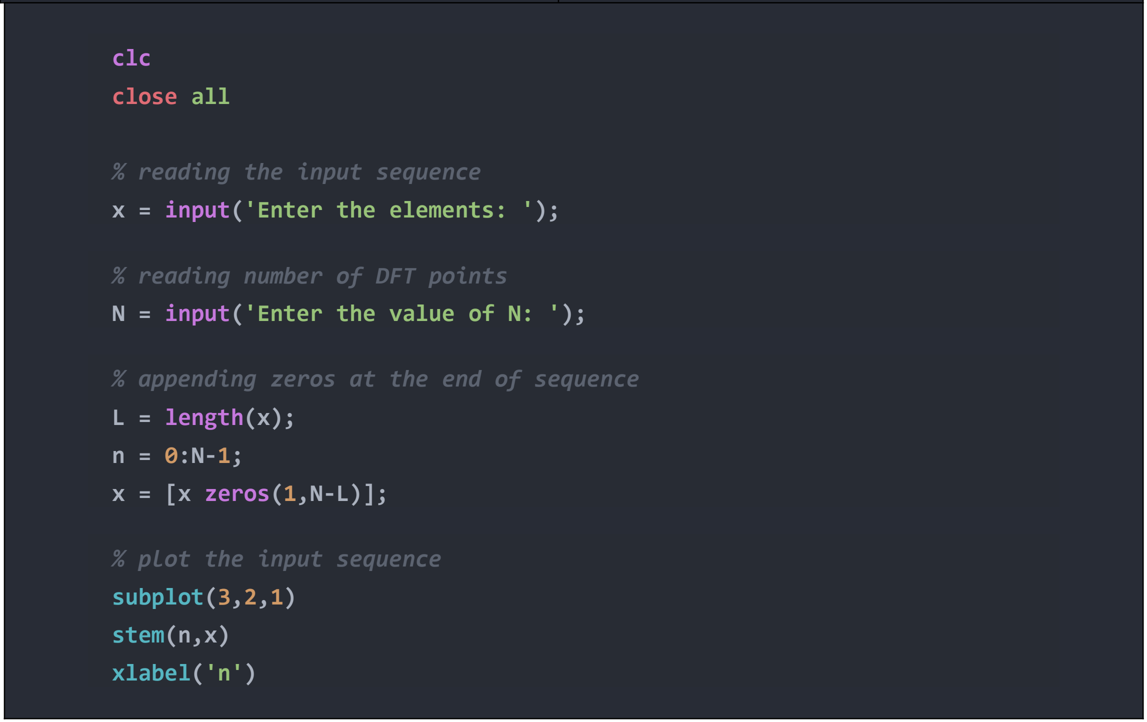


```
function Xk=dft_function(xn,N)
L=length(xn);
xn = [xn,zeros(1,N-L)];
for k=0:N-1
    for n=0:N-1
        Wn=exp(-
1i*2*pi*k*n/N);
        X1(k+1,n+1)=Wn;
    end
end
Xk=X1*xn';
```

%idft function.m



```
function Xk=idft_function(y)

N = length(y);
for n = 0:N-1
    for k = 0:N-1
        Wn=exp(1i*2*pi*k*n/N)/N;
        X1(n+1,k+1)=Wn;
    end
end

Xk=X1*y;
```

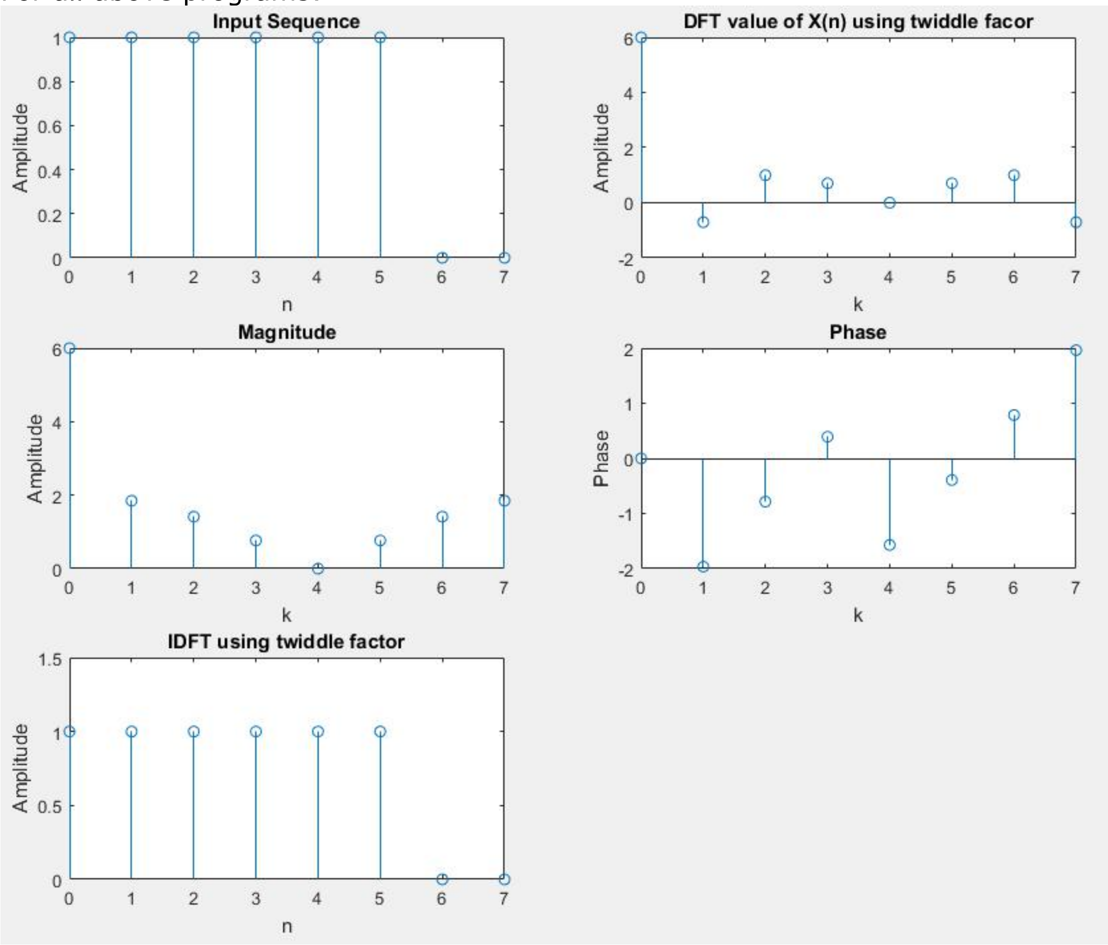


```
clc
close all

% reading the input sequence
x = input('Enter the elements: ');

% reading number of DFT points
N = input('Enter the value of N: ');

% appending zeros at the end of sequence
L = length(x);
n = 0:N-1;
x = [x zeros(1,N-L)];

% plot the input sequence
subplot(3,2,1)
stem(n,x)
xlabel('n')
```

```
ylabel('Amplitude')
title('Input Sequence')

% initializing blank array to fill the IDFT
y = zeros(1,N);

% custom function to find dft
y = dft_function(x,N);
disp('y: ')
disp(y)

% output the dft sequence
k = 0:N-1;
subplot(3,2,2)
stem(k,y)
xlabel('k')
ylabel('Amplitude')
title('DFT value of X(n) using twiddle facor');

magnitude = abs(y);
subplot(3,2,3)
stem(k,magnitude)
xlabel('k')
ylabel('Amplitude')
title('Magnitude')

phase = angle(y);
subplot(3,2,4)
stem(k,phase)
xlabel('k')
ylabel('Phase')
title('Phase')


%IDFT generation using custom function
m=idft_function(y);

% plotting the idft
n = 0:N-1;
subplot(3,2,5)
stem(n,m)
xlabel('n')
ylabel('Amplitude')
title('IDFT using twiddle factor');
```

OUTPUT

For all above programs:



CONCLUSION

Hence, during this practical we have successfully found out DFT and IDFT of given sequence and plotted the output using Matlab. We have also found out DFT and IDFT using in-built function provided by Matlab, and also made a custom function for DFT and IDFT.