# ICS 311 Digital Signal Processing

## Lab 9_ DFT & Convolution

## Name: Abhishek Harsh

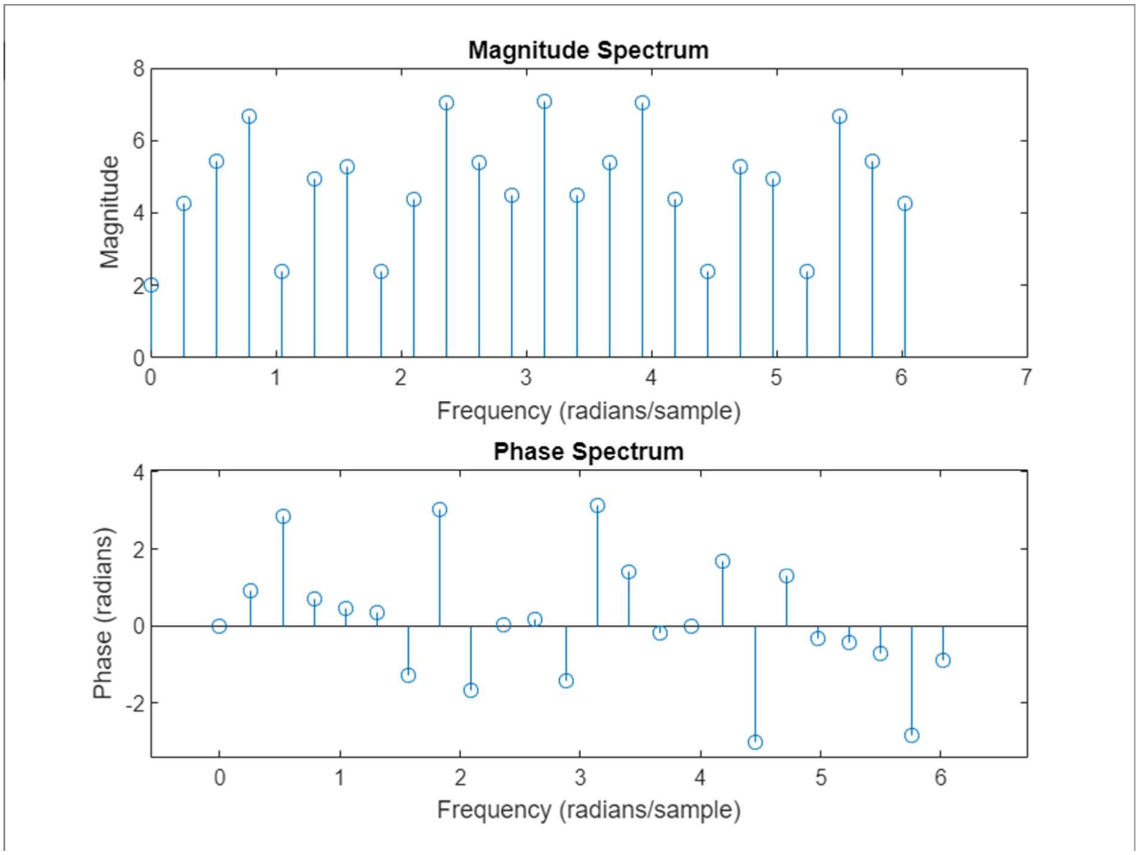
## 2021BCS0036

1. Q 1>Compute the N point DFT of a sequence using Matlab code. Plot its magnitude and phase spectra. (*Choose the value of N and input sequence of your choice*)

## Code:

```
N = 24; % Choose any value for N
sequence = randn(1, N); % Generating a random sequence of length N

% Compute the N-point DFT using MATLAB's fft function
dft_sequence = fft(sequence);

% Calculate the magnitude and phase spectra
magnitud = abs(dft_sequence);
phase_spectrum = angle(dft_sequence);

freq_axis = (0:N-1) * (2 * pi / N); % Normalized

% Plotting the magnitude spectrum
subplot(2, 1, 1);
stem(freq_axis, magnitud);
title('Magnitude Spectrum');
xlabel('Frequency (radians/sample)');
ylabel('Magnitude');

subplot(2, 1, 2);
stem(freq_axis, phase_spectrum);
title('Phase Spectrum');
xlabel('Frequency (radians/sample)');
ylabel('Phase (radians)');
```

## output:



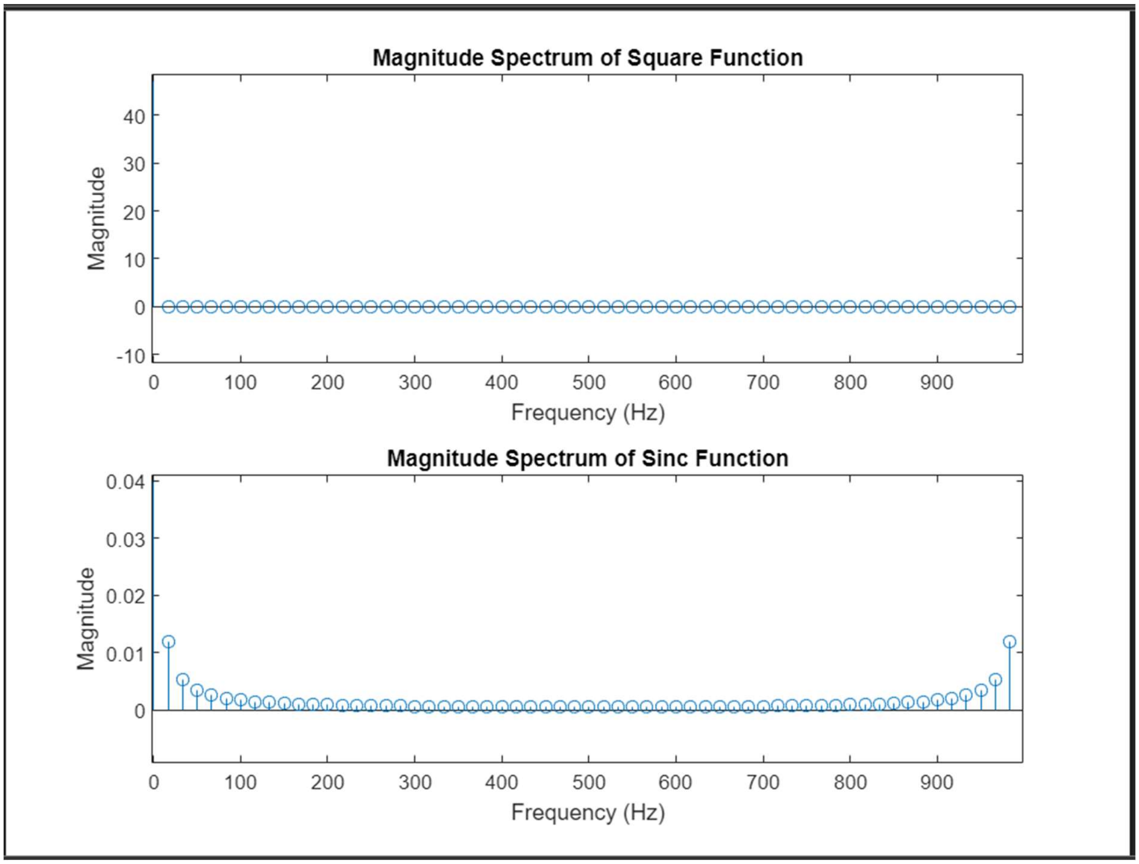
Q2>Perform Matlab experiment for DFT computation of square and sinc function.

## Code:

```
%Abhishek harsh 2021BCS0036
N = 200;
Fs = 1000; % Sampl freq
t = (0:N-1) / Fs; % Time axis


square_wave = square(2 * pi * 5 * t); % Creating a square wave


sinc_function = sinc(10 * (t - (N-1) / 2));

dft_square = fft(square_wave);
dft_sinc = fft(sinc_function);

% magnitude spectra
magnitude_square = abs(dft_square);
magnitude_sinc = abs(dft_sinc);
```

```
frequency_axis = (0:N-1) * (Fs / N);

% Plotting the DFT magnitude spectra
figure;

subplot(2, 1, 1);
stem(frequency_axis, magnitude_square);
title('Magnitude Spectrum of Square Function');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(2, 1, 2);
stem(frequency_axis, magnitude_sinc);
title('Magnitude Spectrum of Sinc Function');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

output:



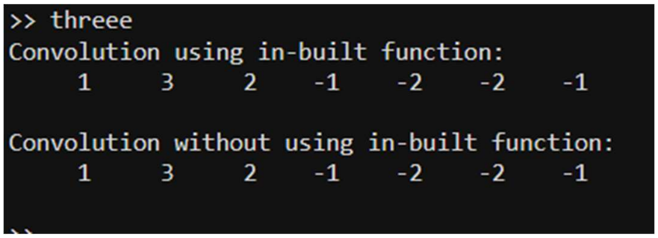
Q3>Write and execute a program in Matlab to find the convolution of the given two sequences with and without in-built functions.

- x *(n)* = [1, 2, 1, 1]
- h *(n)* = [1, 1, -1, -1]

## Code:

```
% Given sequences
%Abhishek harsh 2021BCS0036
x = [1, 2, 1, 1];
h = [1, 1, -1, -1];

% Convolution using in-built function conv
convolution_builtin = conv(x, h);

% Convolution without using in-built function
N = length(x) + length(h) - 1;
convolution_manual = zeros(1, N);

for n = 1:N
    convolution_manual(n) = 0;
    for k = 1:length(x)
        if n - k + 1 > 0 && n - k + 1 <= length(h)
            convolution_manual(n) = convolution_manual(n) + x(k) * h(n - k + 1);
        end
    end
end

% Displaying the results
disp('Convolution using in-built function:');
disp(convolution_builtin);

disp('Convolution without using in-built function:');
disp(convolution_manual);
```

## Output:



```
>> threee
Convolution using in-built function:
     1     3     2    -1    -2    -2    -1

Convolution without using in-built function:
     1     3     2    -1    -2    -2    -1
```

Q4>For the given two sequences in the following, determine the convolution *y*(*n*) = *x*(*n*)**h*(*n*) using with and without in built function in Matlab. Plot all three signals

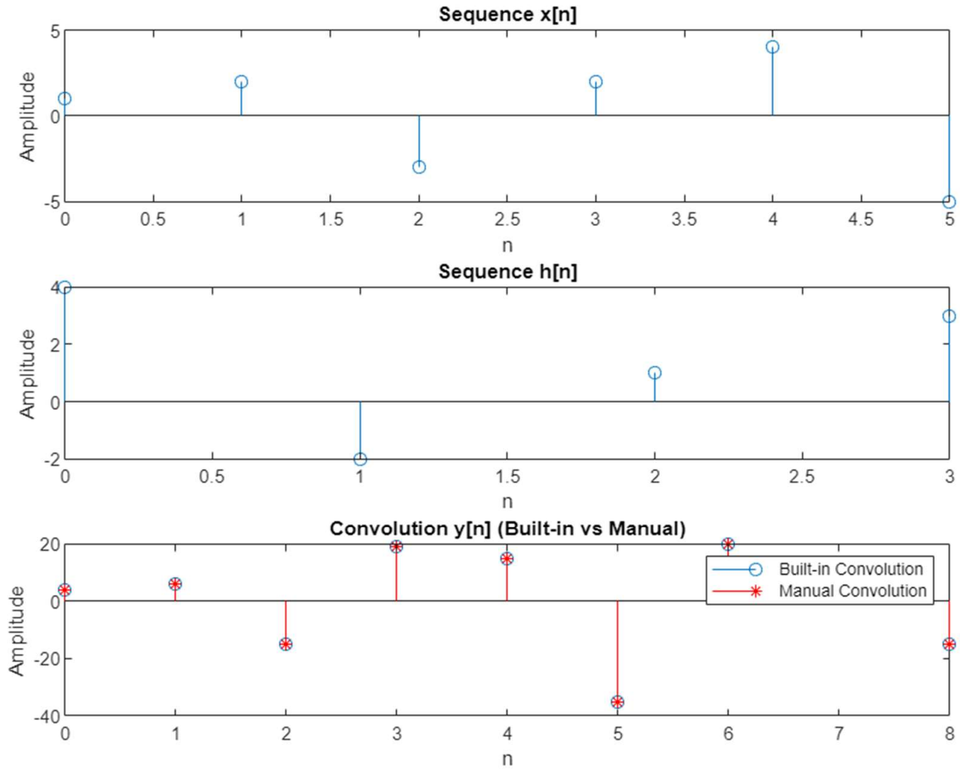
- *x*(*n*) = [ 1  2 -3  2  4 -5]; *h*(*n*) = [ 4 -2  1  3]

## Code:

```
% Given sequences
%Abhishek harsh 2021BCS0036
x = [1, 2, -3, 2, 4, -5];
h = [4, -2, 1, 3];

% Convolution using in-built function conv
y_builtin = conv(x, h);

% Lengths of sequences
N_x = length(x);
N_h = length(h);
N_y = N_x + N_h - 1;
% Convolution without using in-built function
y_manual = zeros(1, N_y);

for n = 1:N_y
    y_manual(n) = 0;
    for k = max(1, n - N_h + 1):min(n, N_x)
        y_manual(n) = y_manual(n) + x(k) * h(n - k + 1);
    end
end
% Plotting the sequences
n_x = 0:N_x - 1;
n_h = 0:N_h - 1;
n_y = 0:N_y - 1;

figure;

subplot(3, 1, 1);
stem(n_x, x);
title('Sequence x[n]');
xlabel('n');
ylabel('Amplitude');

subplot(3, 1, 2);
stem(n_h, h);
title('Sequence h[n]');
xlabel('n');
ylabel('Amplitude');
```

```
subplot(3, 1, 3);
stem(n_y, y_builtin);
hold on;
stem(n_y, y_manual, 'r*');
hold off;
title('Convolution y[n] (Built-in vs Manual)');
xlabel('n');
ylabel('Amplitude');
legend('Built-in Convolution', 'Manual Convolution');
```

## Output:


Sequence x[n]
Amplitude
n
Sequence h[n]
Amplitude
n
Convolution y[n] (Built-in vs Manual)
Amplitude
n
Built-in Convolution
Manual Convolution