==

==

Dan Otieno. EE 384 -> Spring '24. Classwork 3. Due date: 02/06/24. Credit to Sayan Samanta for assistance with Matlab scripts.

Problem 1a......1

Problem 1b......2

Problem 1c - My DFT. 3

Problem 1c - My Inverse DFT. 4

Problem 2a......5

Problem 2b......6

Problem 2c......7

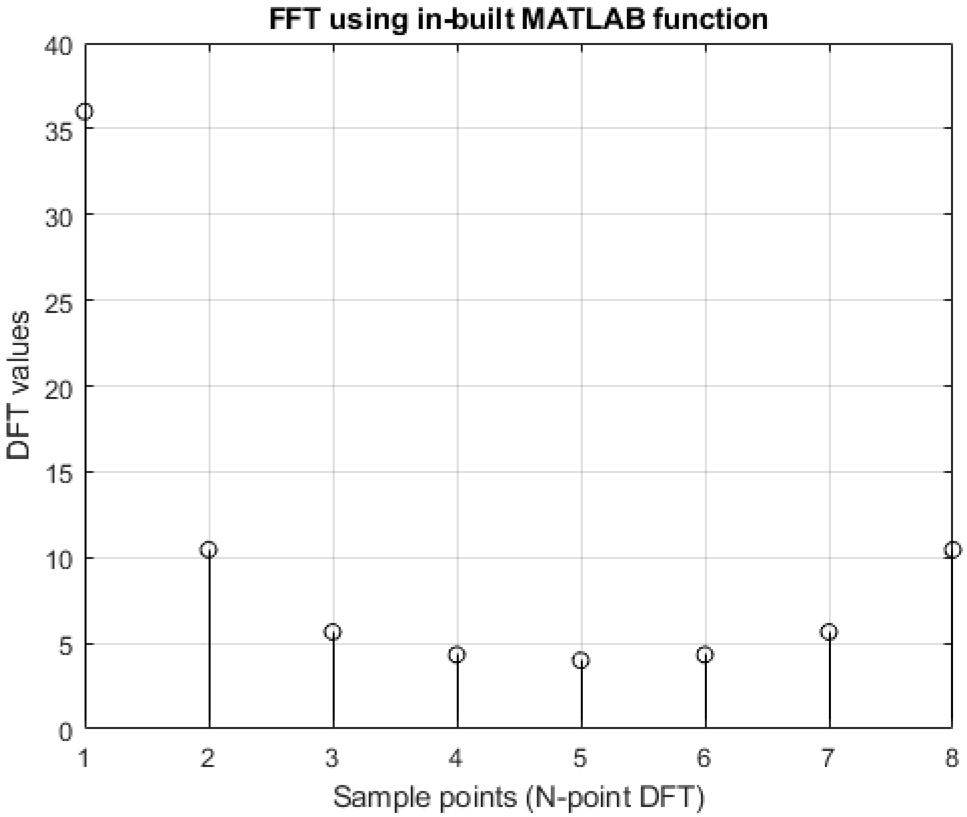
Problem 2d. 8

Functions......9

Problem 1a.

Discrete Fourier Transform and Fast Fourier Transform. a): Use fft function to calculate Fourier Transform. Sequence x = [1, 2, 3, 4, 5, 6, 7, 8].

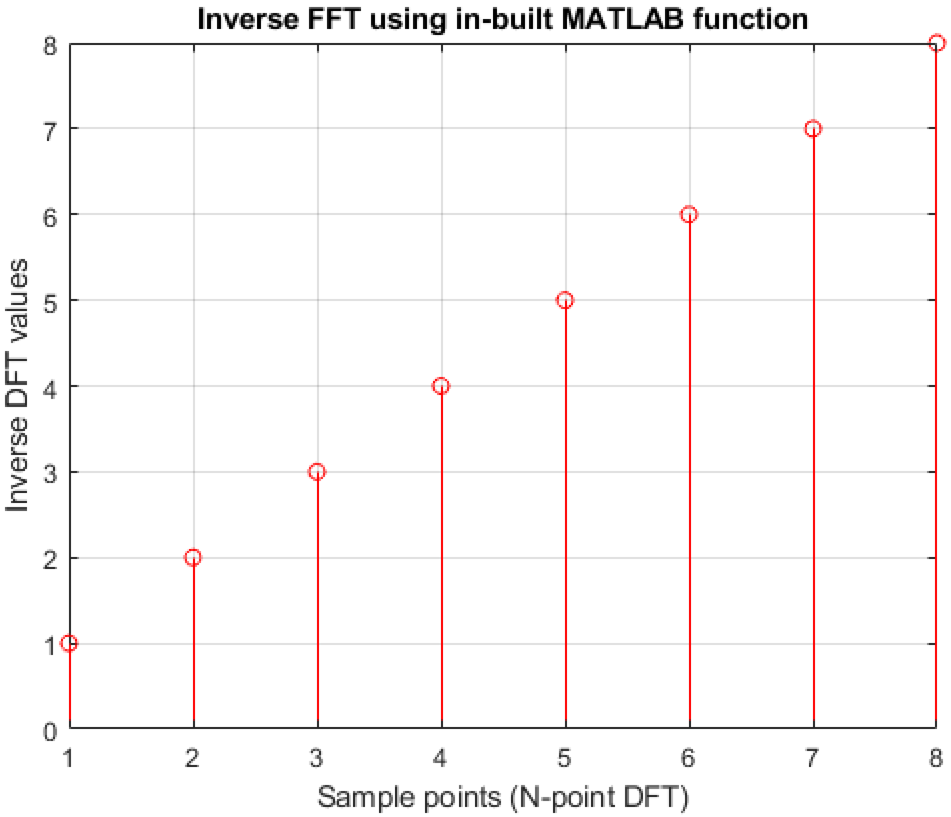
```
close all;clear all;clc
x = [1, 2, 3, 4, 5, 6, 7, 8];
y = fft(x);
figure(1)
stem(abs(y), 'k-');
grid on
title('FFT using in-built MATLAB function');
xlabel('Sample points (N-point DFT)');
ylabel('DFT values');
```



Problem 1b.

Use ifft function to calculate inverse Fourier Transform… ….of the result gotten in (a).

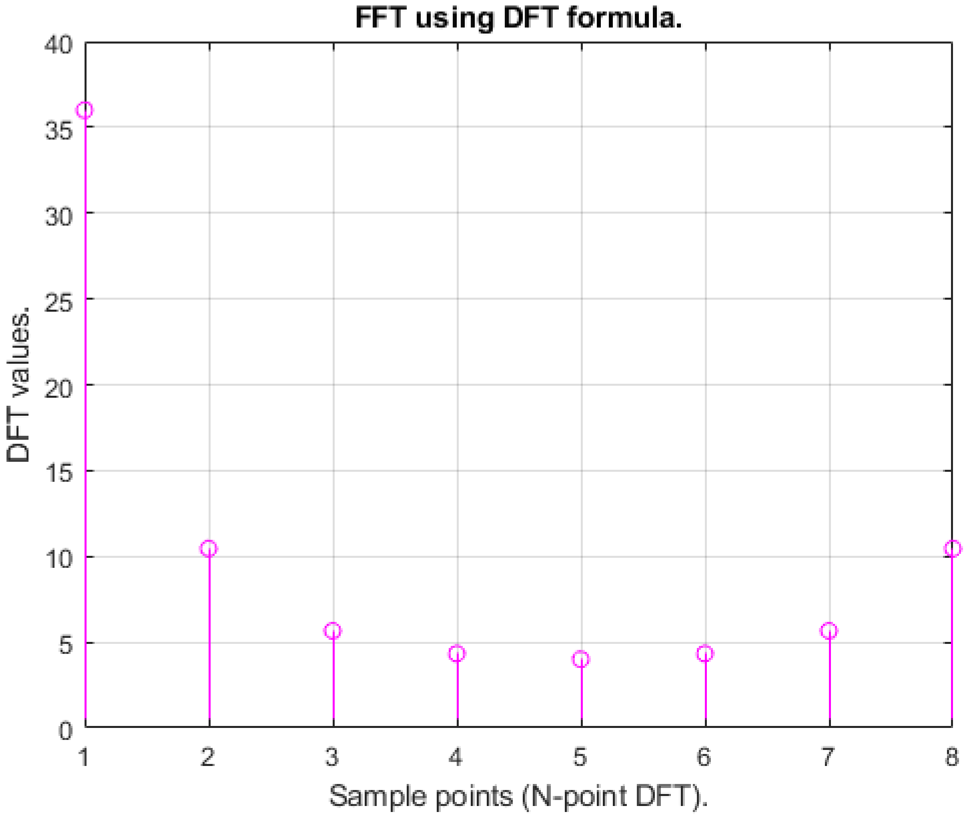
```
z = ifft(y);
figure(2)
stem(abs(z), 'r-');
grid on
title('Inverse FFT using in-built MATLAB function');
xlabel('Sample points (N-point DFT)');
ylabel('Inverse DFT values');
```



Problem 1c - My DFT.

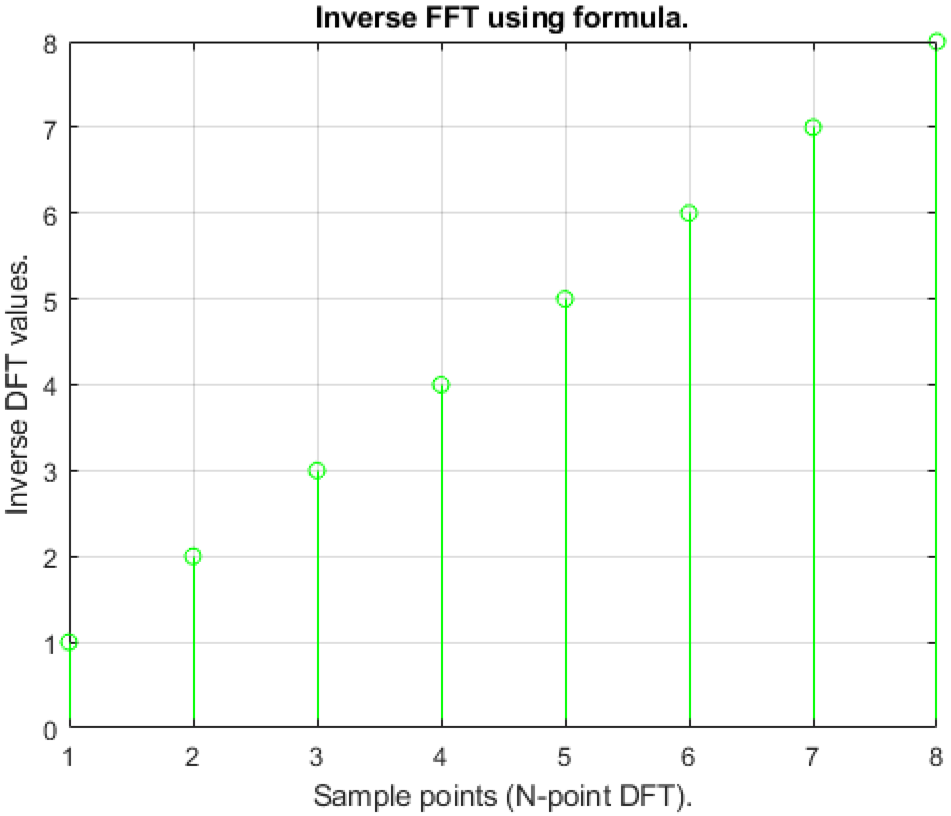
Redo a and b using my own DFT and Inverse DFT functions. .....of the result for (a). Give credit if functions used.

```
X = fnDFT(x);
figure(3)
stem(abs(X), 'm-')
grid on
title('FFT using DFT formula.');
xlabel('Sample points (N-point DFT).');
ylabel('DFT values.');
```



Problem 1c - My Inverse DFT.

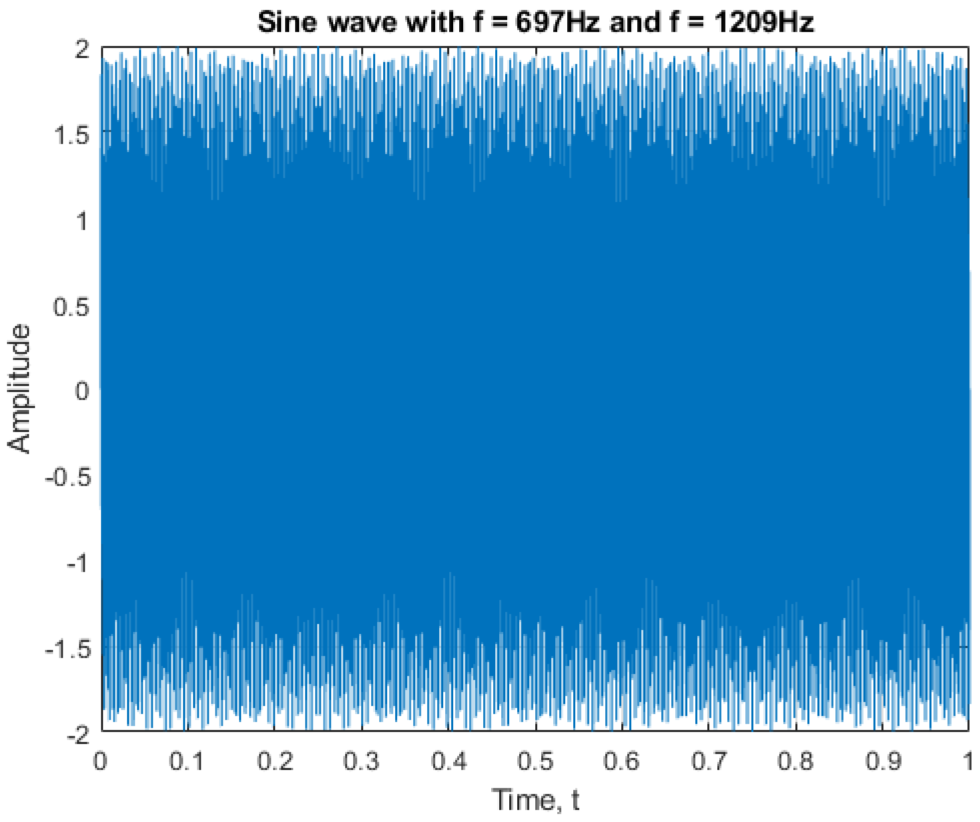
```
Y = fnInvDFT(X);
figure(4)
stem(abs(Y), 'g-')
grid on
title('Inverse FFT using formula.');
xlabel('Sample points (N-point DFT).');
ylabel('Inverse DFT values.');
```



Problem 2a.

Create and plot y(t) = sin(2*pi*f1*t) + sin(2*pi*f2*t).. ...where f1 = 697 Hz & f2 = 1209 Hz.

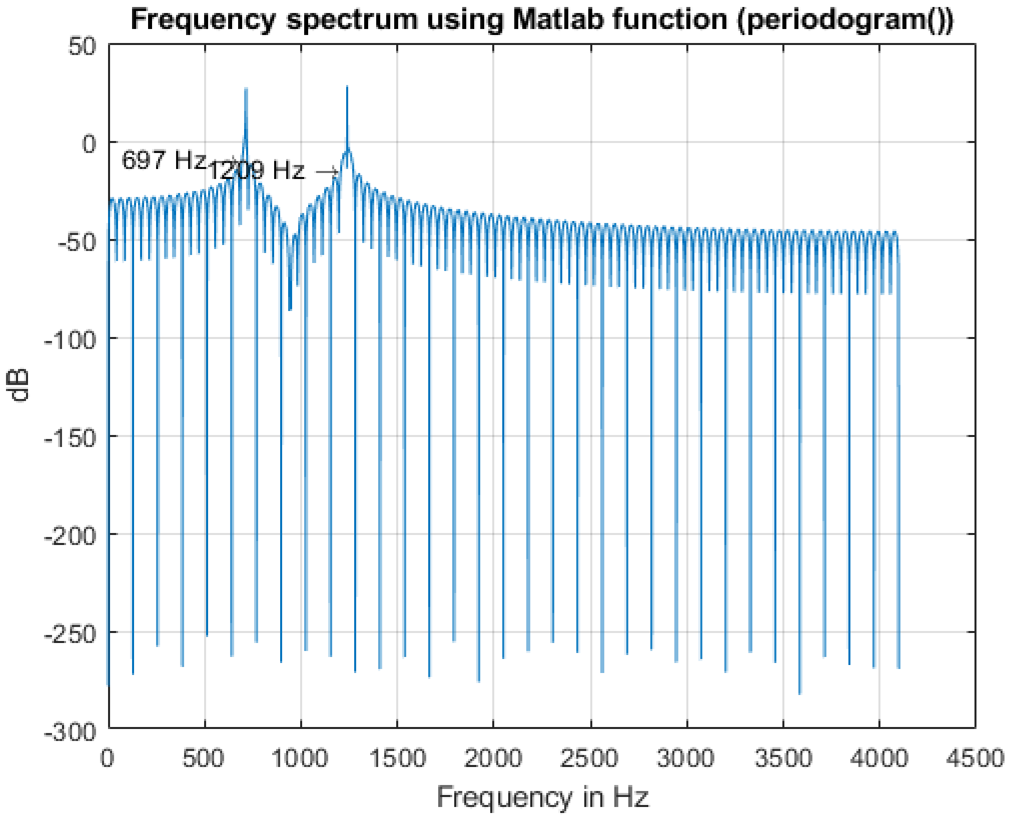
```
fs = 8000;
ts = 1/fs;
t = 0:ts:1;
fa1 = 697;
fa2 = 1209;
y = sin(2*pi*fa1*t) + sin(2*pi*fa2*t);
figure(5);
plot(t,y)
grid on
title(['Sine wave with f = ', num2str(fa1), 'Hz and f = ', num2str(fa2),'Hz'])
xlabel('Time, t'), ylabel('Amplitude')
```



Problem 2b.

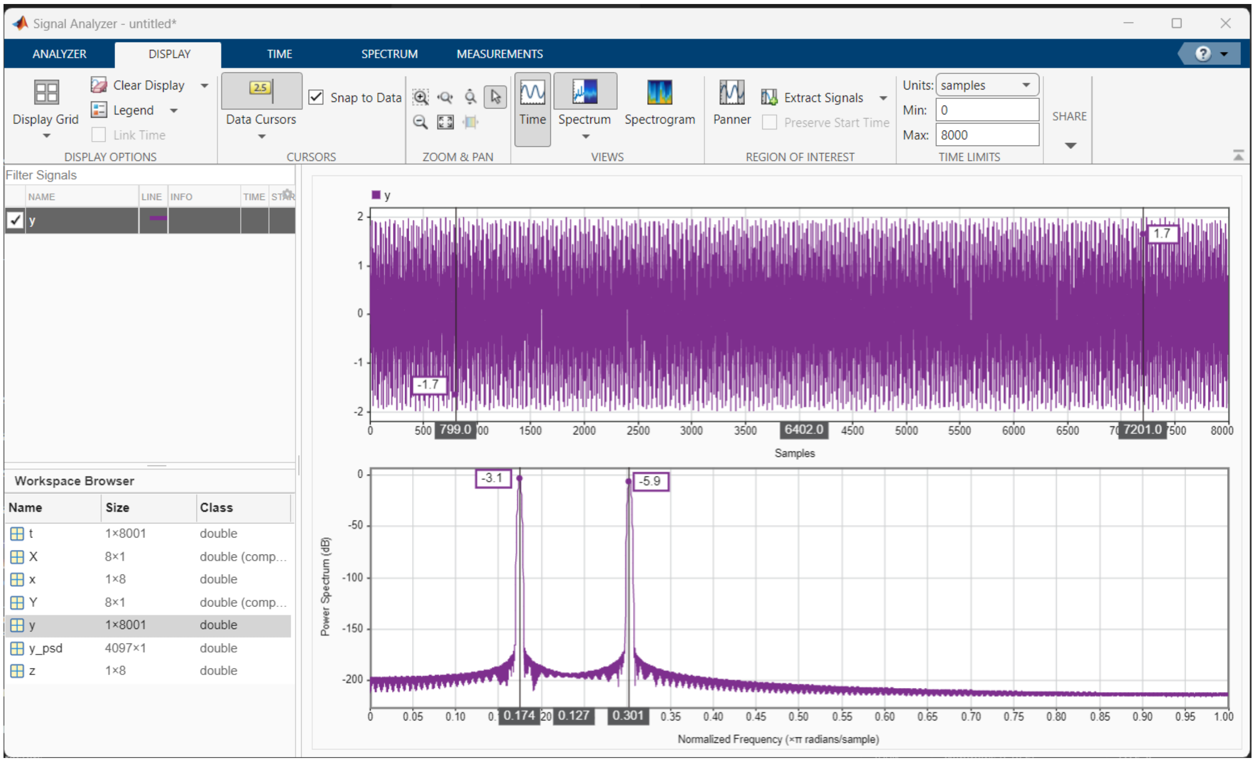
Plot the spectrum of y(t) using spectral estimation tool… …Put markers in the plot to show main frequency components.

```
y_psd = periodogram(y);
figure(6)
plot(10*log10(y_psd));
grid on
title('Frequency spectrum using Matlab function (periodogram())')
xlabel('Frequency in Hz')
ylabel('dB')
text(697, 10*log10(y_psd(697)), '697 Hz \rightarrow', 'HorizontalAlignment','right');
text(1209, 10*log10(y_psd(1209)), '1209 Hz \rightarrow', 'HorizontalAlignment','right');
```



Problem 2c.

Re-do part (b) - Completed using Matlab Signal Analyzer. Image from Signal Analyzer copied and pasted in this document.



☑ Snap to Data

Units: samples
Min: 0
Max: 8000

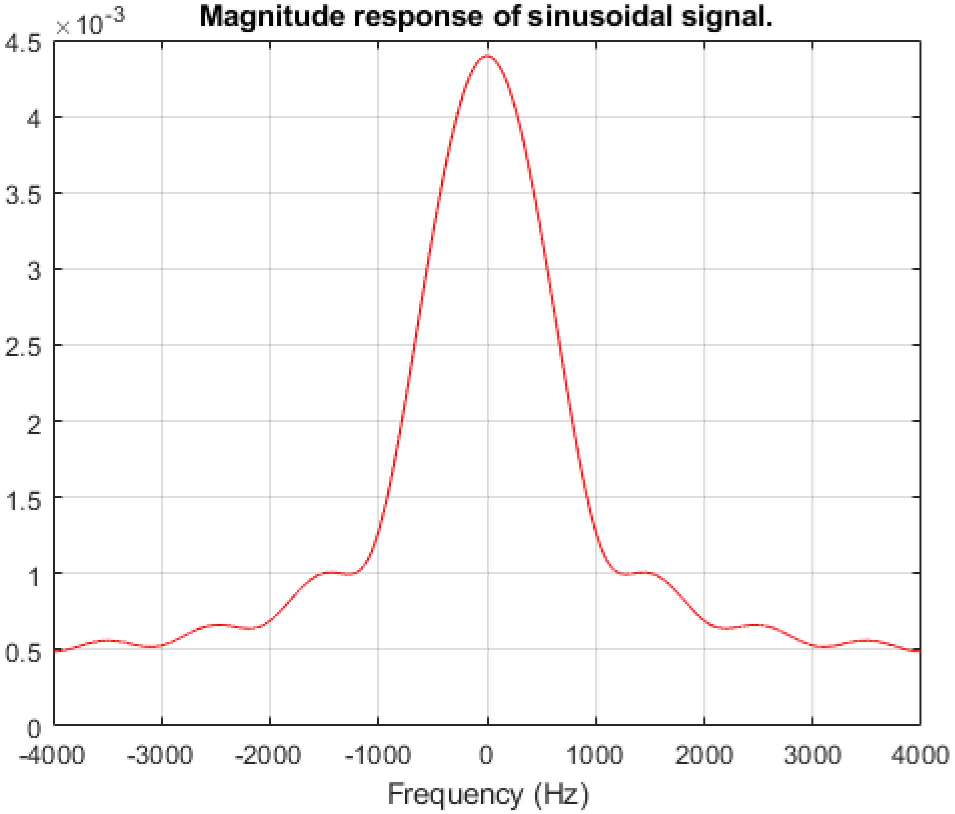
Workspace Browser

Name	Size	Class
t	1×8001	double
X	8×1	double (comp...
x	1×8	double
Y	8×1	double (comp...
y	1×8001	double
y_psd	4097×1	double
z	1×8	double

Problem 2d.

Re-do part (b) using FFT.

```
L = length(y);
n = pow2(nextpow2(L));
y_dft = fft(x, n);
y_s = fftshift(y_dft);
f = (-n/2:n/2-1) * (fs/n);
figure(7)
plot(f, abs(y_s)/n, 'r-');
grid on
xlabel('Frequency (Hz)')
title('Magnitude response of sinusoidal signal.')
```



Functions.

```
function [y] = fnDFT(x)
    N = length(x);
    y = zeros(N, 1);
    for k = 1:N
        X(k+1) = 0;
        for n = 0:N-1
            X(k+1) = X(k+1) + (x(n+1) * exp(-1i*(2*pi/N)*(k-1)*n));
        end
        y(k) = X(k+1);
    end
end

function [x] = fnInvDFT(y)
    N = length(y);
    x = zeros(N, 1);
    for k = 1:N
        Y(k+1) = 0;
        for n = 0:N-1
             Y(k+1) = Y(k+1) + (y(n+1) * exp(-1i*(2*pi/N)*-(k-1)*n));
        end
        x(k)=(1/N)*Y(k+1);
    end
end
```