

L9.1 - Create dft.m function

```
function [X] = dft(x, N)
%DFT Summary of this function goes here
% function to compute the discrete fourier transform of an inputted
% sequence
%   x = sequence input
%   N = the length of the sequence
%   X = the corefficients of the discrete fourir transform
% create variable to hold the DFT output
X = zeros(1, N);
% compute the DFT
for k = 0: N-1 % simulating the summation
    for n = 0: N-1
        X(k+1) = X(k+1) + x(n+1) * exp(-1j * 2 * pi * k * n / N);
    end
end
%finally normalize the output by dividing by N
X = X / N;
end
```

L9.2 - Create the idft.m function

```
function [x] = idft(X, N)
%IDFT Summary of this function goes here
%   computes the inverse discrete fourier transform of an inputted seq.
% initialize x which will hold the output
x = zeros(1, N)

% compute the idft, similar to dft, only the conjugate
for n = 0: N - 1
    for k = 0: N-1
        x(n+1) = x(n+1) + X(k+1) * exp(1j * 2 * pi * k * n / N);
    end
end
%finally normalize the output by dividing by N
x = x / N;
end
```