Lab 1

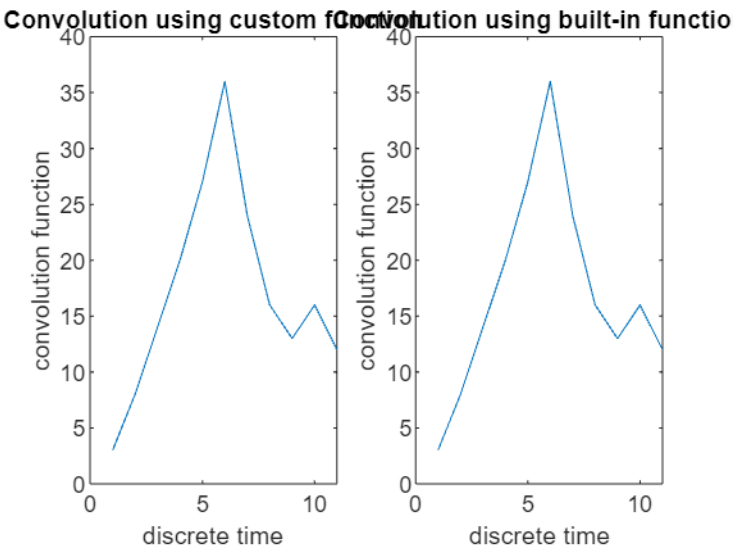
Submitted by:

Saurabh Kumar

SC22B146

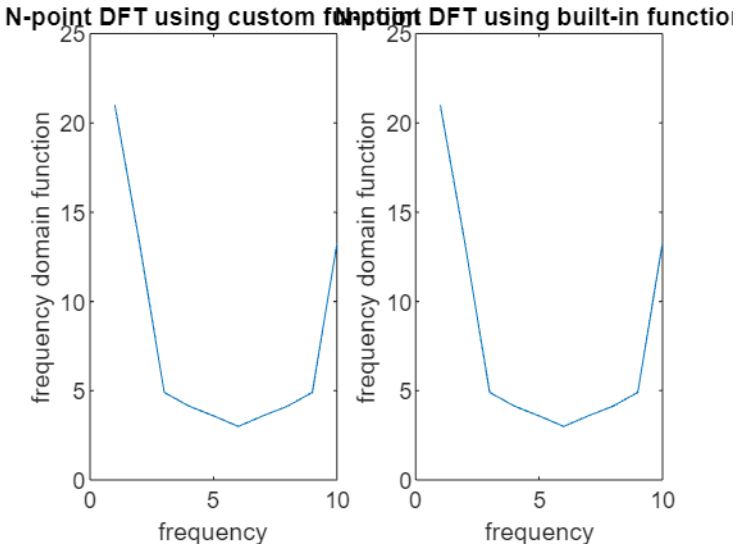
1. Convolution

```
x = [1 2 3 4 5 6];
y = [3 2 1 0 1 2];

conv_xy = convolution(x,y);

figure;
subplot(1,2,1);
plot(conv_xy);
title("Convolution using custom function");
xlabel("discrete time");
ylabel("convolution function");

subplot(1,2,2);
plot(conv(x,y));
title("Convolution using built-in function");
xlabel("discrete time");
ylabel("convolution function");
```



2. a. Discrete Fourier Transform

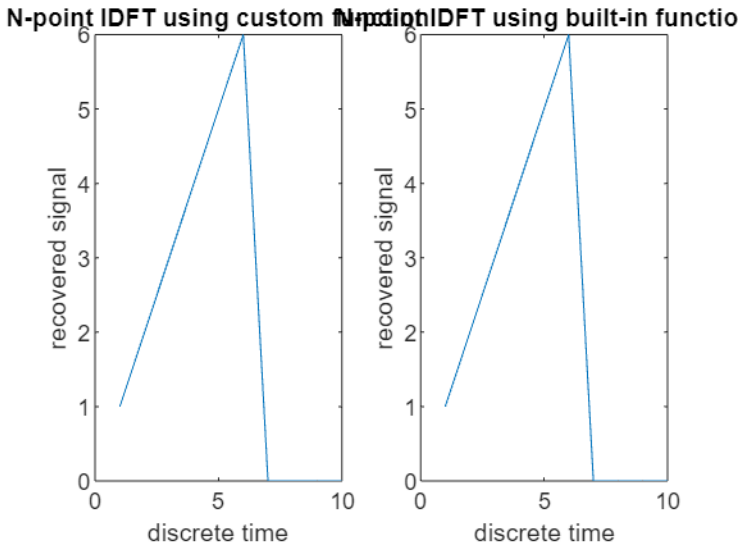
```
x = [1 2 3 4 5 6];

dft_xy = discrete_ft(x,10);
figure;
subplot(1,2,1);
plot(abs(dft_xy));
title("N-point DFT using custom function");
xlabel("frequency");
ylabel("frequency domain function");

subplot(1,2,2)
dft_inbuilt = fft(x,10);
plot(abs(dft_inbuilt));
title("N-point DFT using built-in function");
xlabel("frequency");
ylabel("frequency domain function");
```



2. b. Inverse Discrete Fourier Transform

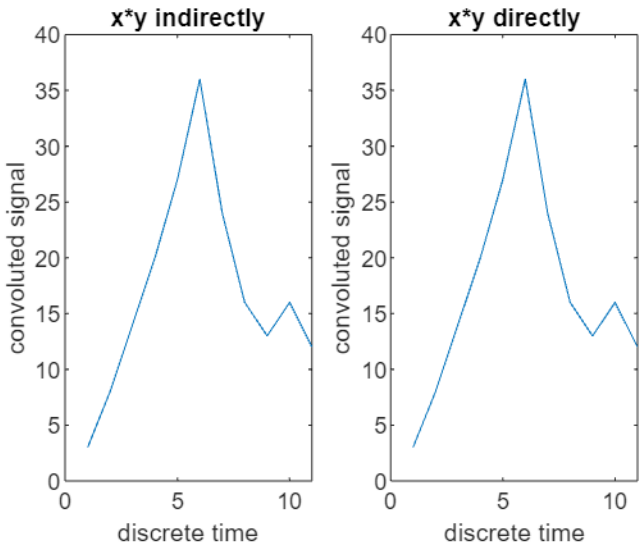
```
idft_xy = discrete_ift(dft_inbuilt,10);
figure;
subplot(1,2,1);
plot(abs(idft_xy));
title("N-point IDFT using custom function");
```

```
xlabel("discrete time");
ylabel("recovered signal");

subplot(1,2,2);
plot(ifft((dft_inbuilt),10));
title("N-point IDFT using built-in function");
xlabel("discrete time");
ylabel("recovered signal");
```



2. c. Indirect Fourier Transform

```
x_conv_y_indirect =
abs(discrete_ift(discrete_ft(x,11).*discrete_ft(y,11),11));
figure;
subplot(1,2,1);
plot(x_conv_y_indirect);
title("x*y indirectly");
xlabel("discrete time");
ylabel("convoluted signal");

x_conv_y_direct = convolution(x,y);
subplot(1,2,2);
plot(x_conv_y_direct);
title("x*y directly");
xlabel("discrete time");
ylabel("convoluted signal");
```



Functions:

Convolution function:

```
function conv_result = convolution(x,y)
    conv_result = zeros(1,length(x)+length(y)-1);

    for k = 1:length(x)+length(y)
        sum = 0;
        for m = 1:length(x)
            if k-m > 0 && k-m <= length(y)
                sum = sum + x(m)*y(k-m);
            end
        end
        if k > 1
            conv_result(k-1) = sum;
        end
    end
end
```

Discrete Fourier Transform function:

```
function dft_result = discrete_ft(x,N)
    dft_result = zeros(1,N);

    if N > size(x)
```

```
        x = [x zeros(1,N-length(x))]; % zero padding to ensure correct
size of x
    end

    for k = 0:N-1
        for n = 0:N-1
            dft_result(k+1) = dft_result(k+1) +
x(n+1)*exp(-1i*2*pi*n*k/N);
        end
    end
end
```

Discrete Inverse Fourier Transform function:

```
function idft_result = discrete_ift(X,N)
    idft_result = zeros(1,N);

    if N > size(X)
        X = [X zeros(1,N-length(X))]; % zero padding to ensure correct
size of X
    end

    for n = 0:N-1
        for k = 0:N-1
            idft_result(n+1) = idft_result(n+1) +
(X(k+1)*exp(1i*2*pi*n*k/N))/N;
        end
    end
end
```