```
clc; clear;
addpath('helpers');



%% Design and Simulation Parameters

% set design parameters
simParams = struct();
simParams.bandwidth          = 100e6;
simParams.sample_rate        = .5e9;
simParams.pulse_width        = 1e-6;

% set simulation paramters
simParams.b                  = simParams.bandwidth / simParams.sample_rate;
simParams.t                  = ((1/simParams.sample_rate):(1/simParams.sample_rate):simParams.pulse_width).';
simParams.nfft               = 2*(simParams.pulse_width*simParams.sample_rate)-1;

% generate spectral template
simParams.spectral_template = windowKaiser(simParams.nfft, 0.585*(1/simParams.b), -80);



%% Generate Waveform

% generate LFM waveform
initialization_waveform = generateLFM(simParams);
plotWaveform(initialization_waveform, simParams.sample_rate);

% generate optimized waveform
waveform = generateOptimizedLFM(simParams, initialization_waveform);
plotWaveform(waveform, simParams.sample_rate);
```