NAICAN
LOPEZ

QUIZ #3

① @ $f_d = \frac{2v f_t}{c}$

$v = 300 m/s$
$f_t = 1GHz$
$c = 3 \times 10^8 m/s$

$f_d = \frac{(2)(300)(1 \times 10^9)}{(3 \times 10^8)}$

$f_d = 2000Hz$

ⓑ $\Delta f = \frac{1}{T_{record}}$

$T_{record} = \frac{1}{\Delta f} = \frac{1}{f_d} = \frac{1}{2000Hz}$

$T_{record} = (5 \times 10^{-4})$

$T_{record} = 5.0 milliseconds$

ⓒ $T_{sample} = \frac{1}{(2 \times 10^9)}$

$T_{sample} = 0.5 nanosce$

$\#of\ samp = \frac{(0.5 \times 10^{-3})}{10.5 \times 10^{-9})}$

$\#of\ samp = 1 \times 10^6 samples$

I personally don't think that it is practical since then there would be one million samples. Unless you have a software that could do it for you.

② @ $\Delta t = 2R$

$f(t) = f_0 e^{kt}$

$f_r(t) = f_0 + k(t - \Delta t)$

$\Delta f = f_r(t) - f(t)$

$\Delta f = k(t - \Delta t) - kt$

$\Delta f = -k\Delta t$

$\Delta f = -\frac{k\,2R}{c}$ can be

$R = \frac{c\Delta f}{2k}$

ⓑ $\Delta f = 25MHz$
$k = 1MHz/\mu s$
$c = 300m/\mu s$

$R = \frac{(3 \times 10^8)(25 \times 10^6)}{(2)(1 \times 10^6)}$

$R = 3.75 \times 10^9 m / 3.750 \times 10^6 km / 3750 km$

③ Google doc

④ $= 75ns (7.5e^{-8}s)$

② Octave

③ $\Delta f = \frac{2Vr}{\lambda} \Big/ R = \frac{c\Delta f}{2}$

$\Delta t = 60ns$
$= 6e^{-8}s$
$c = 3 \times 10^8 m/s$

$R = \frac{c\,\Delta t}{2}$

$R = \frac{(3 \times 10^8)(6 \times 10^{-8})}{2}$

$R = 9m$

1 Echo: delay 60ns; 2 Echo: delay 120ns

$R_2 = \frac{(c)(120 \times 10^{-9})}{2}$

$R_2 = 18m$

② Linear Image Processing
Octave (Google doc)

③ Bonus
Octave (google doc)

3.

```
data = [
 5.00000000e-09 1.02061118e+00
 1.00000000e-08 9.79391642e-01
 1.50000000e-08 8.67605029e-01
 2.00000000e-08 9.56968413e-01
 2.50000000e-08 6.14329409e-01
 3.00000000e-08 6.97675879e-01
 3.50000000e-08 4.08610321e-01
 4.00000000e-08 3.82217034e-01
 4.50000000e-08 3.09705366e-01
 5.00000000e-08 1.44454093e-01
 5.50000000e-08 -1.77190519e-01
 6.00000000e-08 -3.75209303e-01
 6.50000000e-08 -3.80431524e-01
 7.00000000e-08 -6.42008513e-01
 7.50000000e-08 -7.95983173e-01
 8.00000000e-08 -9.05569007e-01
 8.50000000e-08 -8.31614567e-01
 9.00000000e-08 -7.27390471e-01
 9.50000000e-08 -9.79623756e-01
 1.00000000e-07 -1.01739513e+00
 1.05000000e-07 -7.87257708e-01
 1.10000000e-07 -8.03526296e-01
 1.15000000e-07 -7.70464801e-01
 1.20000000e-07 -6.49764403e-01
 1.25000000e-07 -6.63797841e-01
 1.30000000e-07 -6.44535253e-01
 1.35000000e-07 -4.49225391e-01
 1.40000000e-07 -1.71300471e-01
 1.45000000e-07 2.02456570e-01
 1.50000000e-07 9.56703152e-02
 1.55000000e-07 4.18932743e-01
 1.60000000e-07 5.23747539e-01
 1.65000000e-07 5.35025885e-01
 1.70000000e-07 5.52559344e-01
 1.75000000e-07 8.73421591e-01
];

t = data(:,1);
v = data(:,2);
Fs = 1 / mean(diff(t));
figure;
specgram(v, 16, Fs);
```

```
title('Spectrogram of Voltage Signal');
xlabel('Time (s)');
ylabel('Frequency (Hz)');
colorbar;
```

4)
2.

```
[s, f, t_spec] = specgram(v, 64, Fs);
[~, idx] = min(abs(t_spec - 7.5e-8));
s_power = abs(s(:, idx)).^2;
[peak_vals, locs] = findpeaks(s_power, 'MinPeakHeight', max(s_power)/5);
frequencies_MHz = f(locs) / 1e6;
Delta_f_MHz = max(frequencies_MHz) - min(frequencies_MHz);
```

2
The number should be 2017, I think it's a Chengdu J-20.

```
img = imread('aircraft_image.jpg');
if size(img, 3) == 3
img = rgb2gray(img);
img = im2double(img);
gaussian_kernel = [1 2 1;
                   2 4 2;
                   1 2 1] / 16;
laplacian_kernel = [0 -1  0;
                   -1  4 -1;
                    0 -1  0];

smoothed_img = conv2(img, gaussian_kernel, 'same');
edges = conv2(smoothed_img, laplacian_kernel, 'same');
sharpened_img = smoothed_img + edges;
figure;
subplot(1,3,1), imshow(img), title('Original Image');
subplot(1,3,2), imshow(smoothed_img), title('Smoothed Image');
subplot(1,3,3), imshow(sharpened_img), title('Sharpened Image');
```

3. Bonus
a

```
Fs = 44100;
T = 1;
t = linspace(0, T, Fs*T);
f0_1 = 1000; f1_1 = 4000;
f0_2 = 1500; f1_2 = 4500;
```

```
chirp1 = chirp(t, f0_1, T, f1_1, 'linear');
chirp2 = chirp(t, f0_2, T, f1_2, 'linear');
signal = chirp1 + chirp2;
```

b

```
noisy_signal = signal + 0.3 * randn(size(signal));
pkg load signal;
[b, a] = butter(4, [800 5000]/(Fs/2), 'bandpass');
filtered_signal = filter(b, a, noisy_signal);
```

c

```
sound(filtered_signal, Fs);
figure;
spectrogram(filtered_signal, 256, 200, 512, Fs, 'yaxis');
title('Spectrogram of Filtered Signal');
```

I think they would interfere.