DFT AND IDFT COMPUTATIONS

LAB # 09

CSE402L Digital Signal Processing Lab

Submitted by: **Shah Raza**

Registration No: **18PWCSE1658**

Class Section: **B**

"On my honor, as a student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: ______________

Submitted to: **Engr. Faiz Ullah**

Friday, February 19th, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

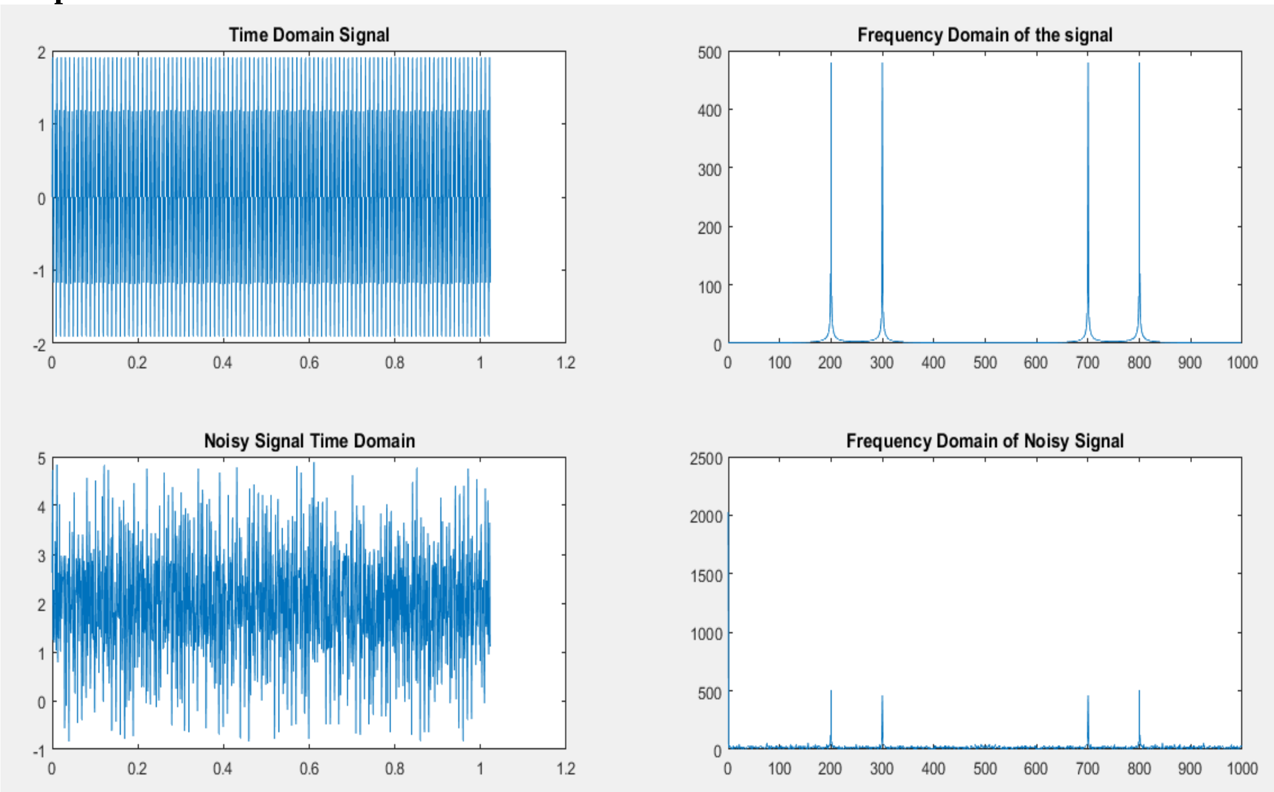
Practical:

Generate 1024 samples of a signal containing two sinusoids of frequency 200 and 300 Hz, sampled at 1000 Hz. Obtain the frequency response of the resultant signal. Add some noise in the resultant signal and again compute the frequency response. Compare the two plots. Take IFFT of the contaminated signal and compare it to the original signal.

Code:

```
clc
clear all
close all
fs = 1000;
N = 1024;
t = (0:N-1)/fs;
x1 = sin(2*pi*200*t);
x2 = sin(2*pi*300*t);
x = x1+x2;
noise = 1+2*rand(size(t));
xN = x+noise;
xk = fft(x);
xNk = fft(xN);
f = ((0:N-1)*fs)/N;
y1 = ifft(xk);
y2 = ifft(xNk);
subplot(2,2,1)
plot(t,y1);
title('Time Domain Signal');
subplot(2,2,2)
plot(f,abs(xk));
title('Frequency Domain of the signal');
subplot(2,2,3)
plot(t,y2);
title('Noisy Signal Time Domain');
subplot(2,2,4)
plot(f,abs(xNk));
title('Frequency Domain of Noisy Signal');
```

Output:



Questions:

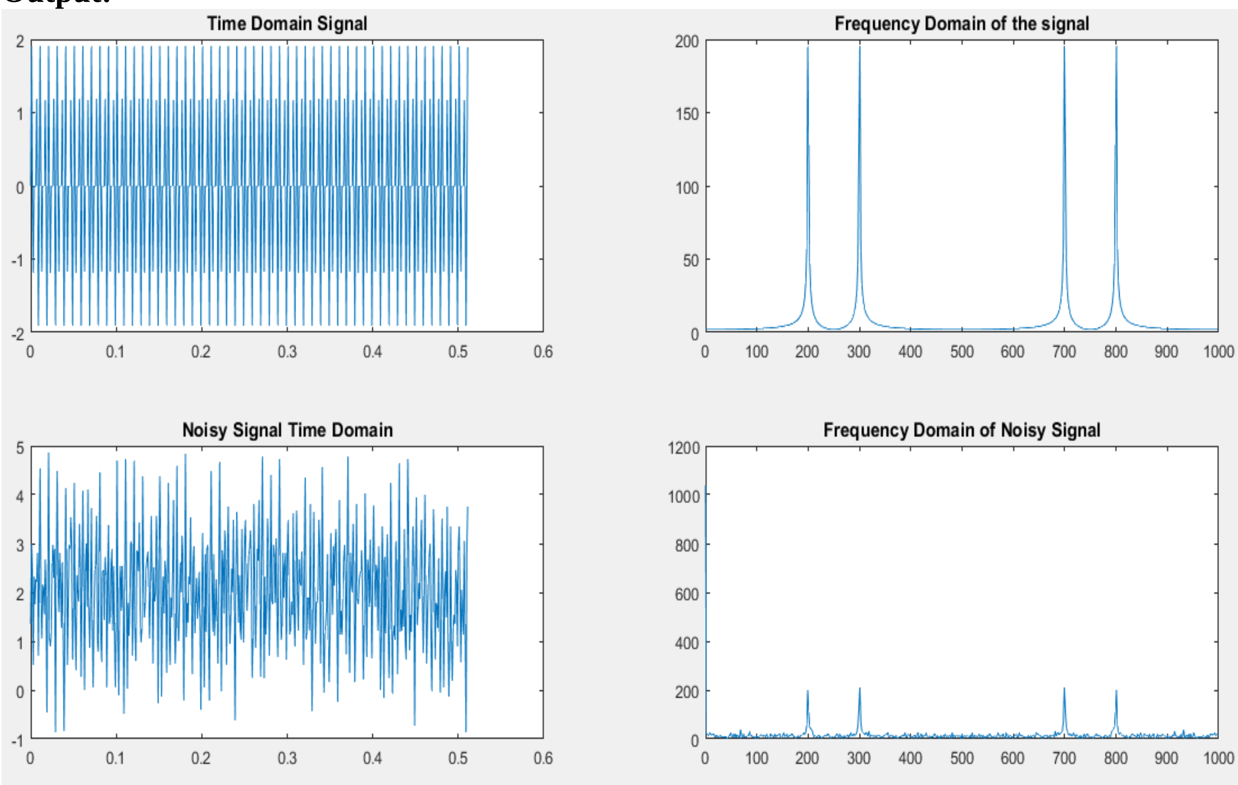
1. **Reduce the number of samples to 512. What happens to the signals? Why?**

Code:

```
clc
clear all
close all
fs = 1000;
N = 512;
t = (0:N-1)/fs;
x1 = sin(2*pi*200*t);
x2 = sin(2*pi*300*t);
x = x1+x2;
noise = 1+2*rand(size(t));
xN = x+noise;
xk = fft(x);
xNk = fft(xN);
f = ((0:N-1)*fs)/N;
y1 = ifft(xk);
y2 = ifft(xNk);
subplot(2,2,1)
plot(t,y1);
title('Time Domain Signal');
subplot(2,2,2)
plot(f,abs(xk));
title('Frequency Domain of the signal');
```

```
subplot(2,2,3)
plot(t,y2);
title('Noisy Signal Time Domain');
subplot(2,2,4)
plot(f,abs(xNk));
title('Frequency Domain of Noisy Signal');
```

Output:



Discussion:

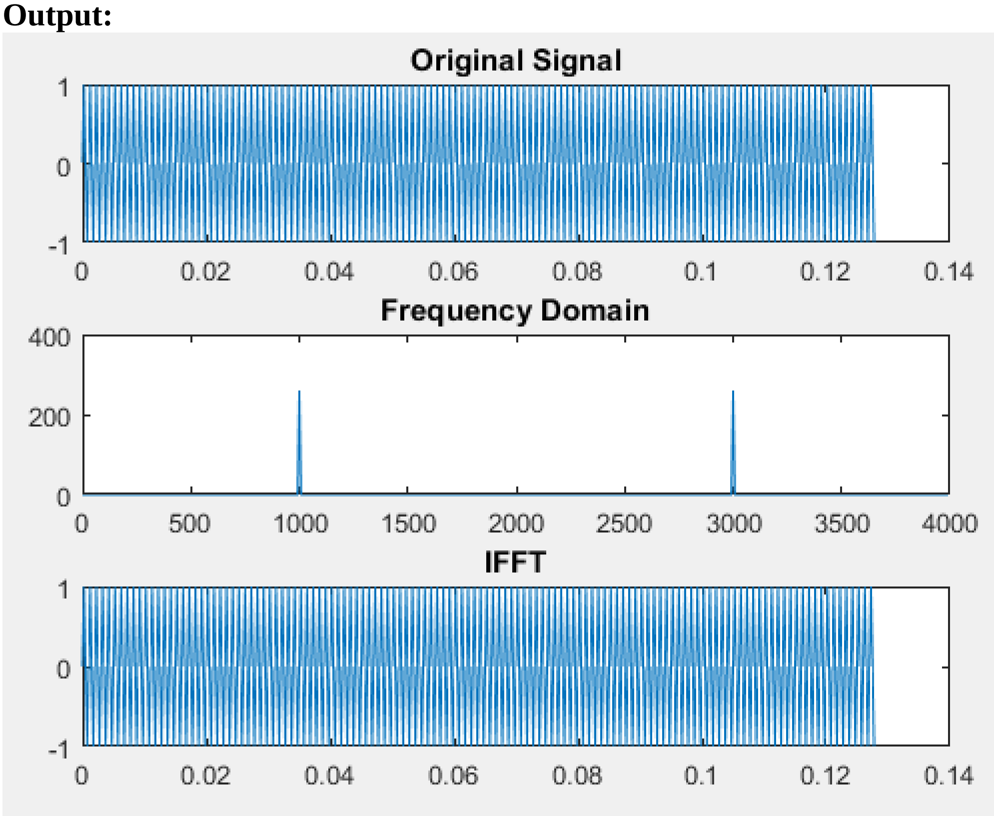
The samples are reduced to 512 from 1024.

2. **Using IFFT, generate 512 samples of a 1000 Hz sinusoid sampled at 4000 Hz.**

Code:

```
clc
clear all
close all
fs = 4000;
N = 512;
t = (0:N-1)/fs;
x = sin(2*pi*1000*t);
xk = fft(x);
f = ((0:N-1)*fs)/N;
y = ifft(xk);
subplot(3,1,1)
plot(t,x);
title('Original Signal');
subplot(3,1,2)
plot(f,abs(xk));
title('Frequency Domain');
subplot(3,1,3)
plot(t,y);
title('IFFT');
```

Output:

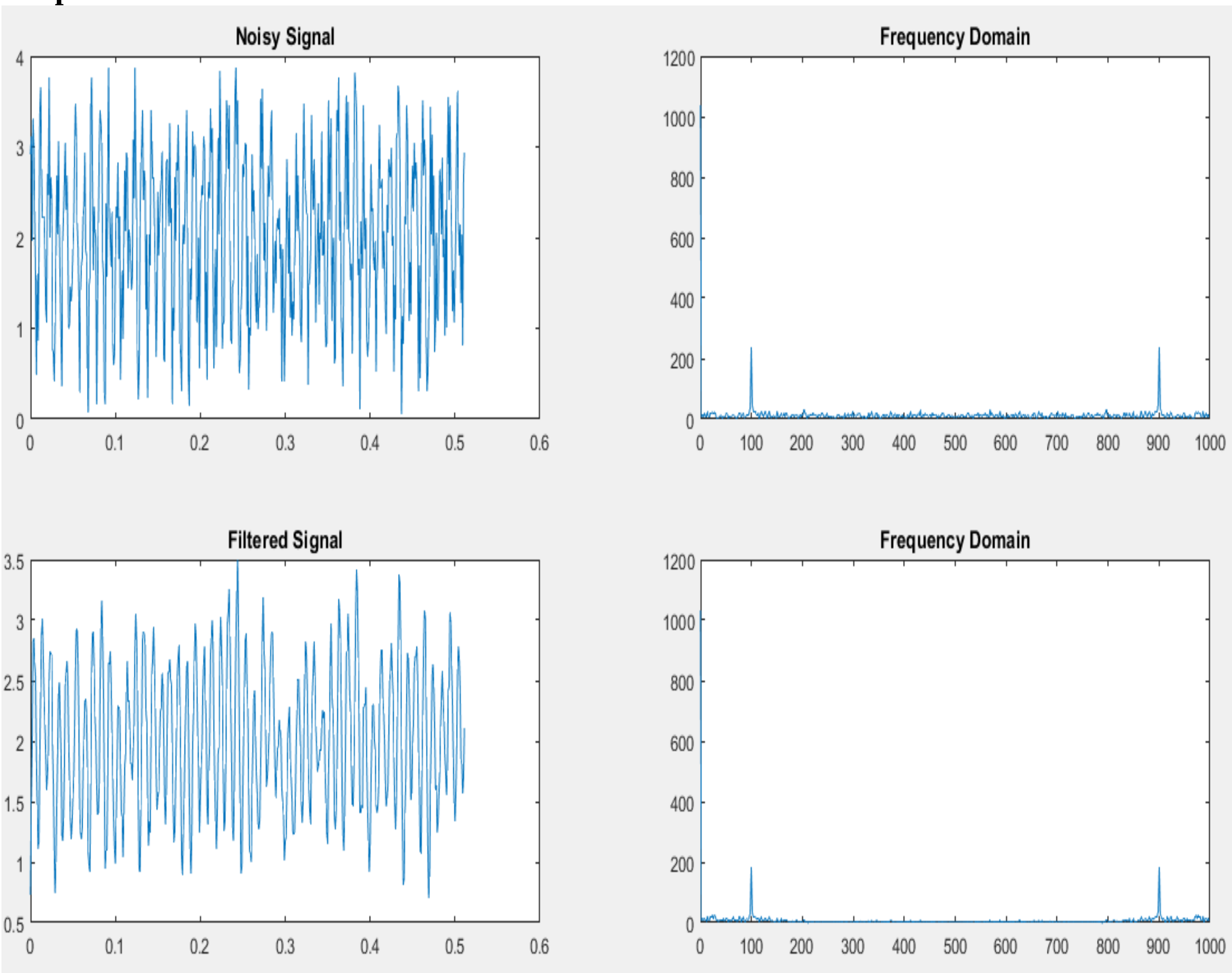


3. Generate a contaminated sinusoid of frequency 100 Hz. Apply this signal to a moving-average filter. Plot the FFT of input to the filter and output of the filter.

Code:

```
clc
clear all
close all
fs = 1000;
N = 512;
t = (0:N-1)/fs;
x = sin(2*pi*100*t);
noise = 1+2*rand(size(t));
xN = x+noise;
xNk = fft(xN);
f = ((0:N-1)*fs)/N;
m = 4;
b = (1/m)*ones(1,m);
y = filter(b,1,xN);
yk = fft(y);
subplot(2,2,1)
plot(t,xN);
title('Noisy Signal');
subplot(2,2,2)
plot(f,abs(xNk));
title('Frequency Domain');
subplot(2,2,3)
plot(t,y);
title('Filtered Signal');
subplot(2,2,4)
plot(f,abs(yk));
title('Frequency Domain');
```

Output:



4. **Generate a signal containing two sinusoids of frequencies 400 Hz and 850 Hz, sampled at 4000 Hz. Plot its FFT for N = 100, 128, 200, 512.**

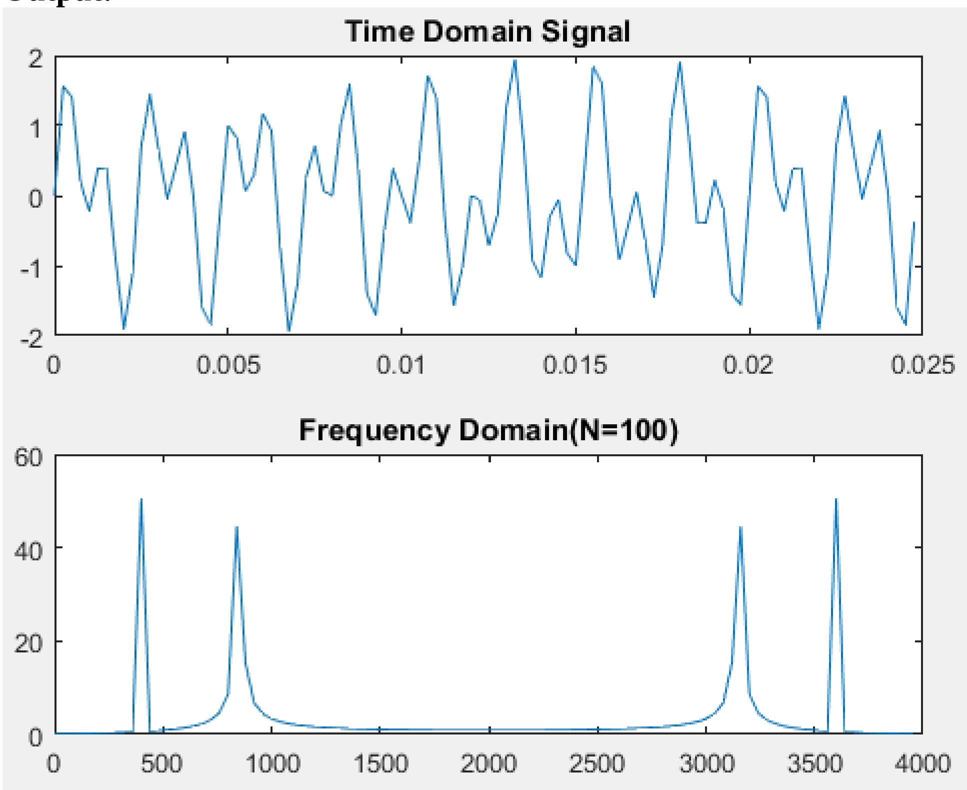
Code:

N=100:

```
clc
clear all
close all
fs = 4000;
N = 100;
t = (0:N-1)/fs;
x1 = sin(2*pi*400*t);
x2 = sin(2*pi*850*t);
x = x1+x2;
xk = fft(x);
f = ((0:N-1)*fs)/N;
subplot(2,1,1)
plot(t,x);
title('Time Domain Signal');
subplot(2,1,2)
plot(f,abs(xk));
```

```
title('Frequency Domain(N=100)');
```

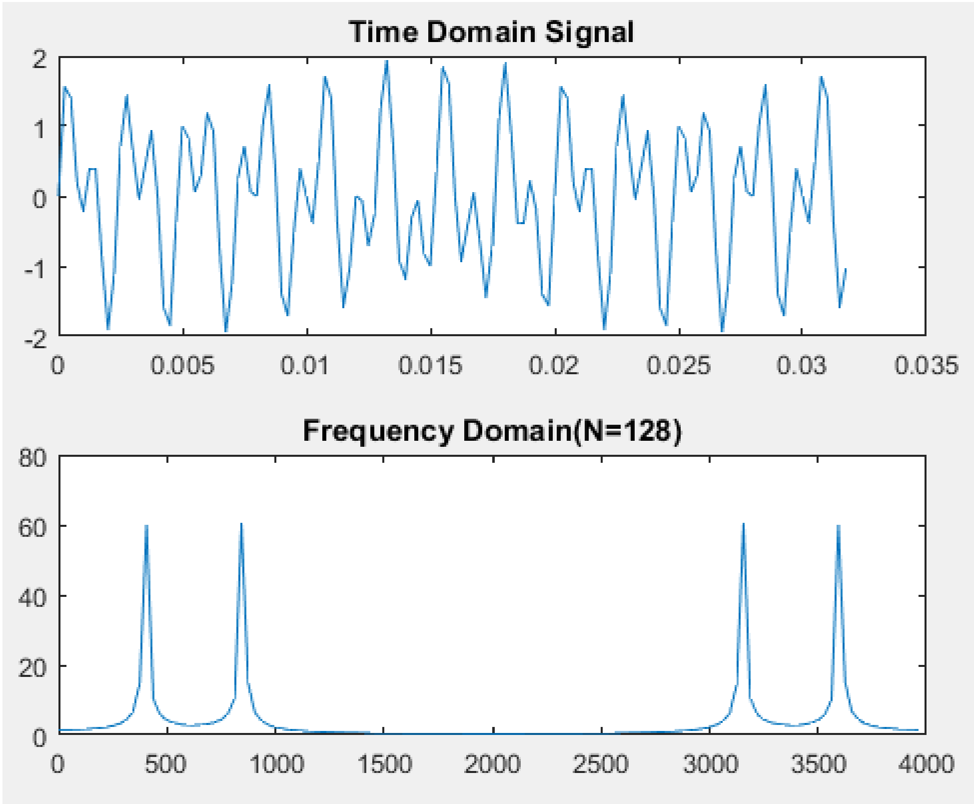
Output:



N=128:

```
clc
clear all
close all
fs = 4000;
N = 128;
t = (0:N-1)/fs;
x1 = sin(2*pi*400*t);
x2 = sin(2*pi*850*t);
x = x1+x2;
xk = fft(x);
f = ((0:N-1)*fs)/N;
subplot(2,1,1)
plot(t,x);
title('Time Domain Signal');
subplot(2,1,2)
plot(f,abs(xk));
title('Frequency Domain(N=128)');
```

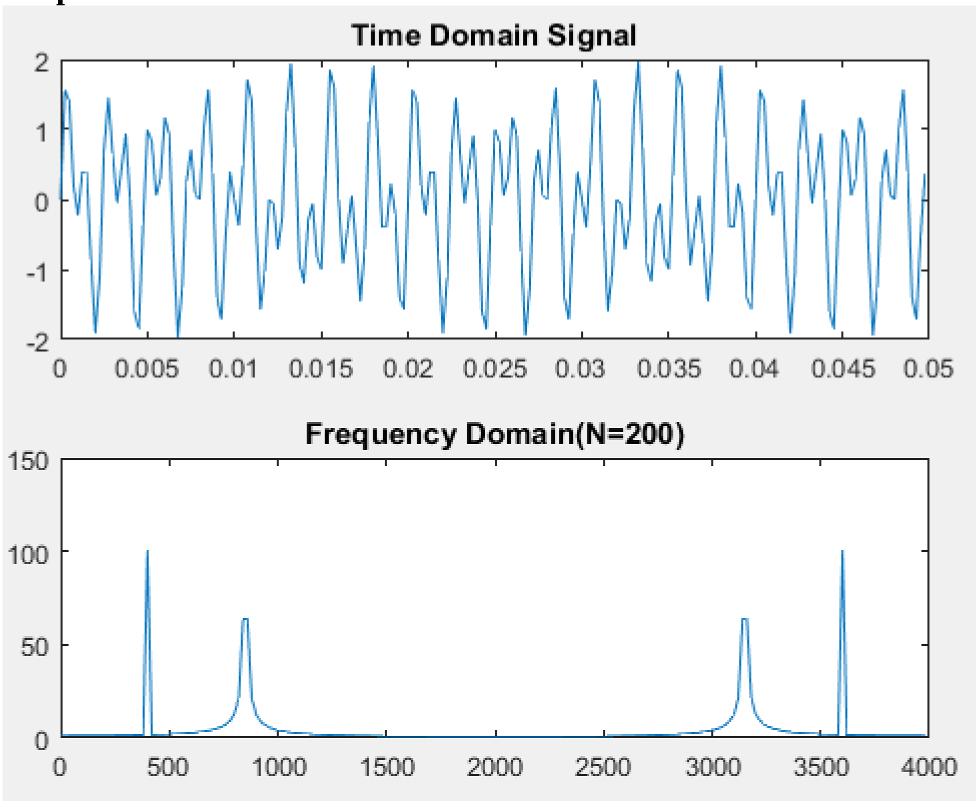
Output:



N=200:

```
clc
clear all
close all
fs = 4000;
N = 200;
t = (0:N-1)/fs;
x1 = sin(2*pi*400*t);
x2 = sin(2*pi*850*t);
x = x1+x2;
xk = fft(x);
f = ((0:N-1)*fs)/N;
subplot(2,1,1)
plot(t,x);
title('Time Domain Signal');
subplot(2,1,2)
plot(f,abs(xk));
title('Frequency Domain(N=200)');
```

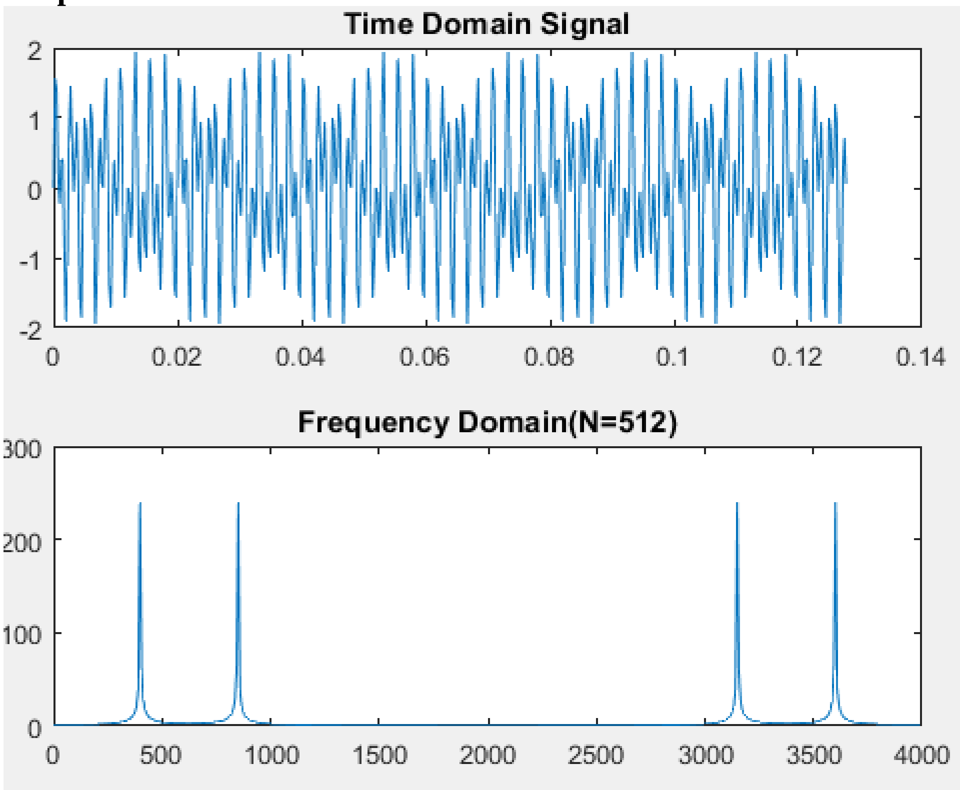
Output:



N=512:

```
clc
clear all
close all
fs = 4000;
N = 512;
t = (0:N-1)/fs;
x1 = sin(2*pi*400*t);
x2 = sin(2*pi*850*t);
x = x1+x2;
xk = fft(x);
f = ((0:N-1)*fs)/N;
subplot(2,1,1)
plot(t,x);
title('Time Domain Signal');
subplot(2,1,2)
plot(f,abs(xk));
title('Frequency Domain(N=512)');
```

Output:

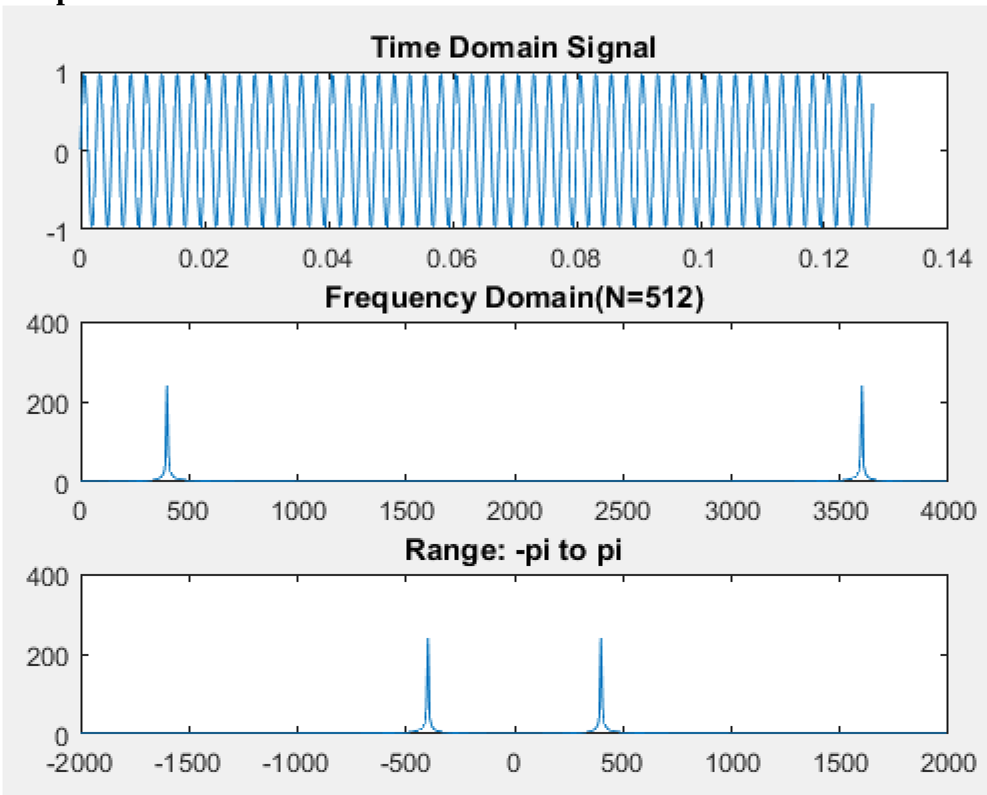


5. **Use fftshift() to represent the frequency in the range of -pi to pi. Use MATLAB's Help for this purpose.**

Code:

```
clc
clear all
close all
fs = 4000;
N = 512;
t = (0:N-1)/fs;
x = sin(2*pi*400*t);
xk = fft(x,N);
f = ((0:N-1)*fs)/N;
subplot(3,1,1)
plot(t,x);
title('Time Domain Signal');
subplot(3,1,2)
plot(f,abs(xk));
title('Frequency Domain(N=512)');
y  = fftshift(xk);
f = ((-N/2:(N-1)/2)*fs)/N;
subplot(3,1,3)
plot(f,abs(y));
title('Range: -pi to pi');
```

Output:


Time Domain Signal
1
0
-1
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
Frequency Domain(N=512)
400
200
0
0
500
1000
1500
2000
2500
3000
3500
4000
Range: -pi to pi
400
200
0
-2000
-1500
-1000
-500
0
500
1000
1500
2000