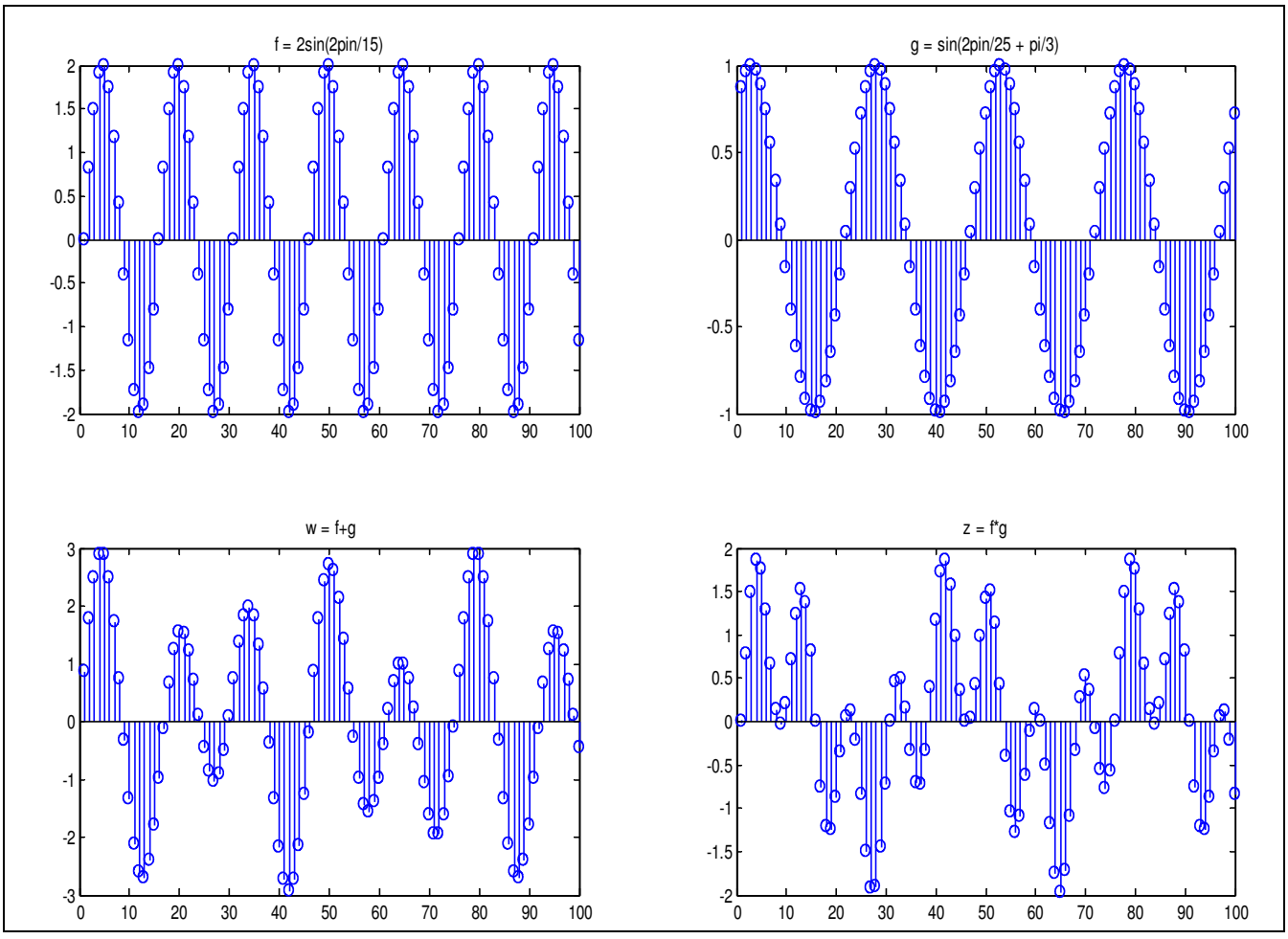
Problem 1

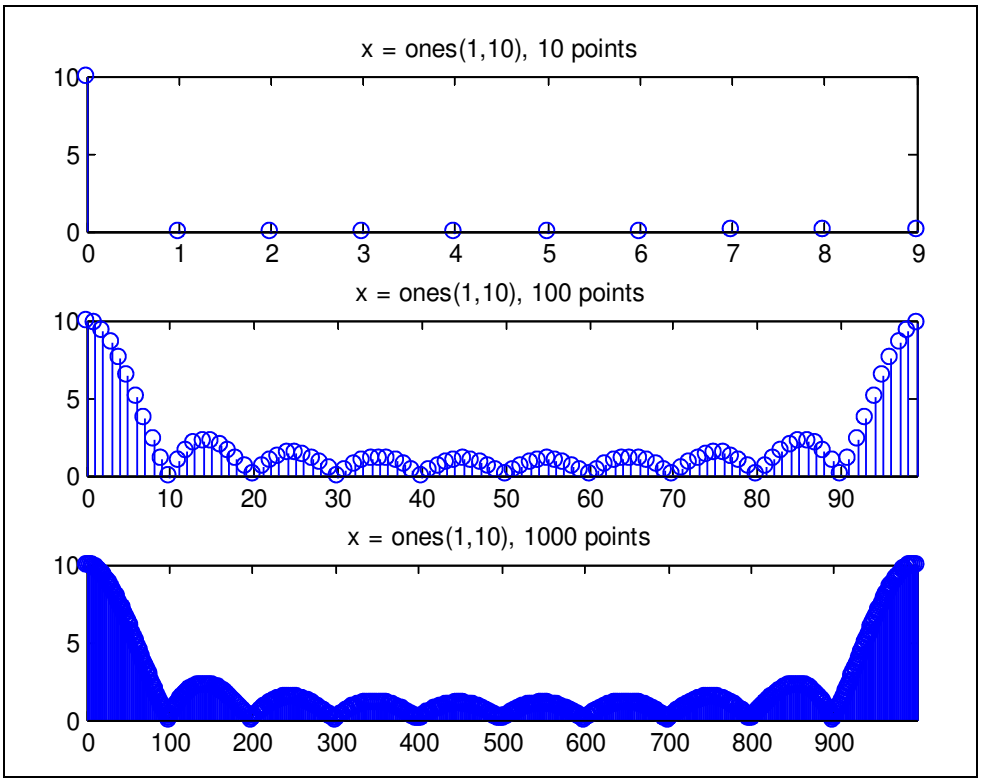
```
function signal = create_sinusoidal(fre, amp, pha, len)
n = 0:len-1;
signal = amp*sin(2*pi*fre*n + pha);
end

f = create_sinusoidal(1/15, 2, 0 ,300);
g = create_sinusoidal(1/25, 1, pi/3 ,300);
w = f+g;
z = f.*g;

subplot(2,2,1);stem(f); title('f = 2sin(2pin/15)')
subplot(2,2,2);stem(g); title('g = sin(2pin/25 + pi/3)')
subplot(2,2,3);stem(w);title(' w = f+g')
subplot(2,2,4);stem(z);title('z = f*g')
```

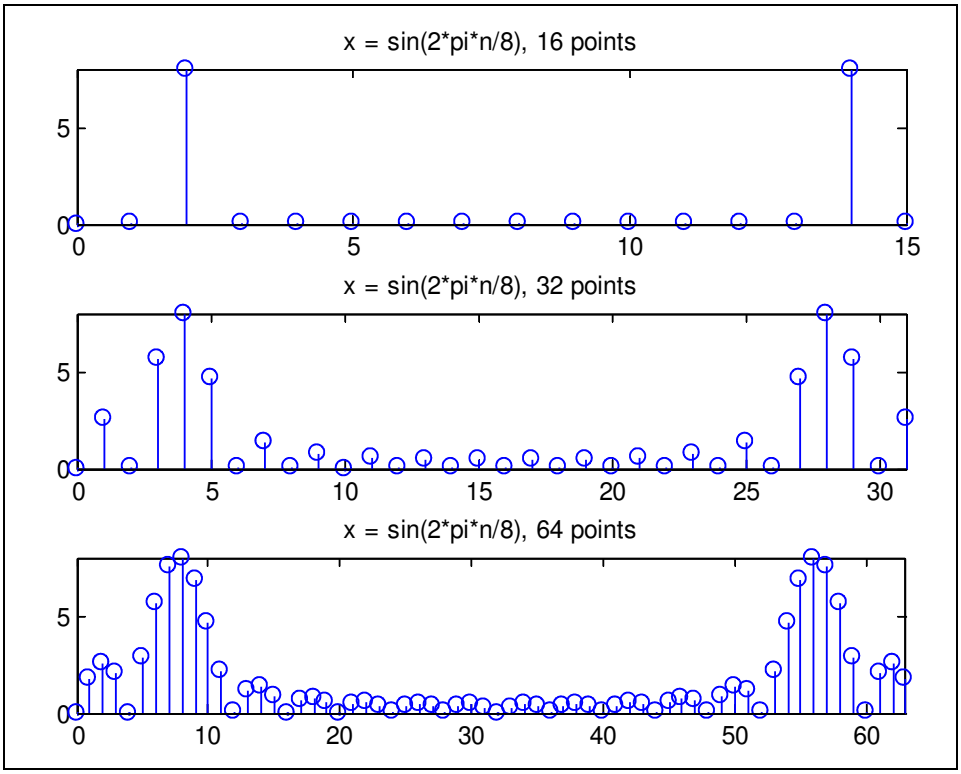
The frequencies of these 4 signals are 15, 25, 75 and 75.

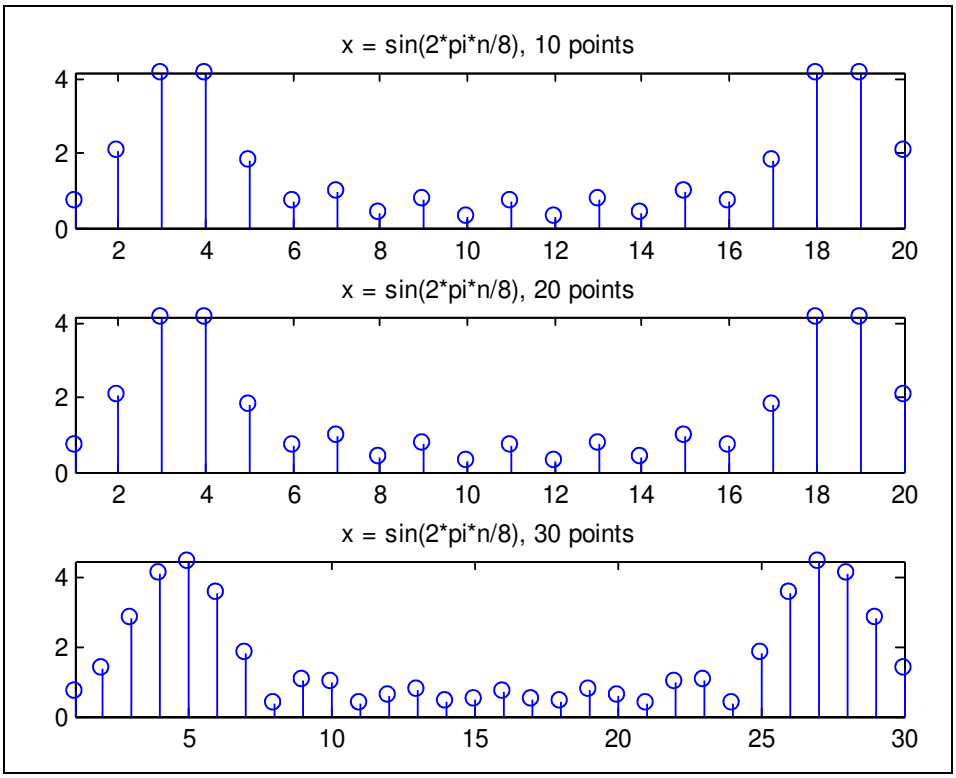


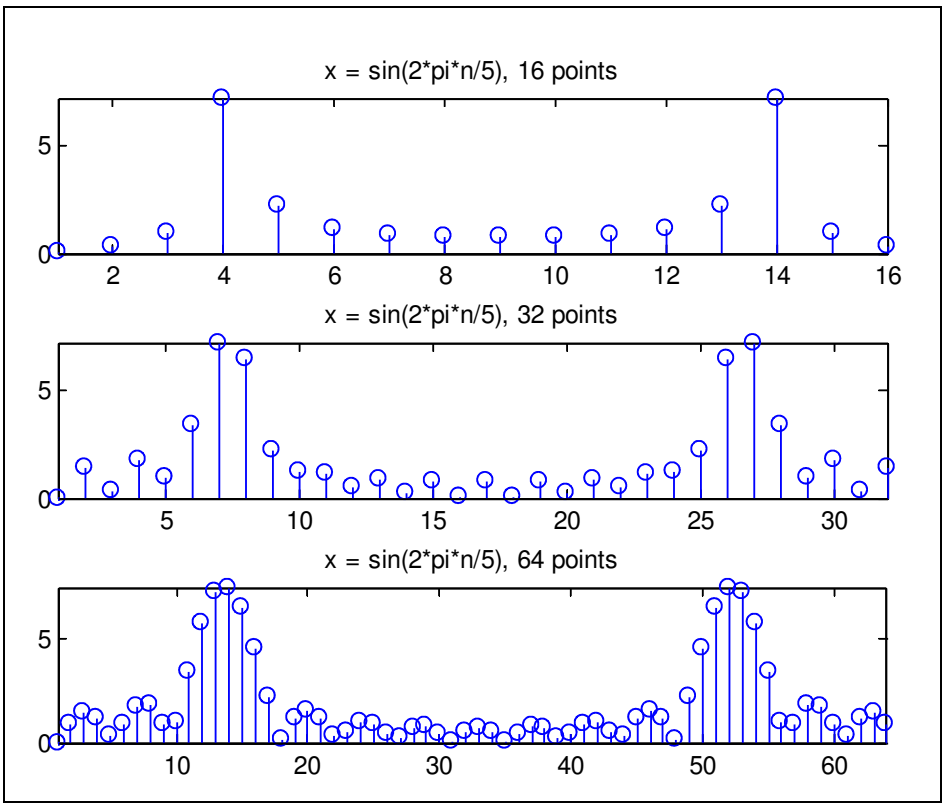
Problem 2.1

```
function [Y,y] = DFT(x,L)
N = length(x);
% zero padding
y = [x, zeros(1,(L-N))];
%     take DFT
Y = zeros(1,L);
n = 0:L-1;
   for k = 0:L-1
        Y(k+1) = sum(y.*exp(-j*2*pi/L*n*k));
   end
end

x = ones(1,10);
[H1,h1] = DFT(x,10);
[H2,h2] = DFT(x,100);
[H3,h3] = DFT(x,1000);

subplot(3,1,1); stem(0:9,abs(H1));title('x = ones(1,10), 10 points');
subplot(3,1,2); stem(0:99,abs(H2));title('x = ones(1,10), 100 points');
subplot(3,1,3); stem(0:999,abs(H3));title('x = ones(1,10), 1000 points');
```




x = sin(2*pi*n/8), 16 points
x = sin(2*pi*n/8), 32 points
x = sin(2*pi*n/8), 64 points


x = sin(2*pi*n/8), 10 points
x = sin(2*pi*n/8), 20 points
x = sin(2*pi*n/8), 30 points



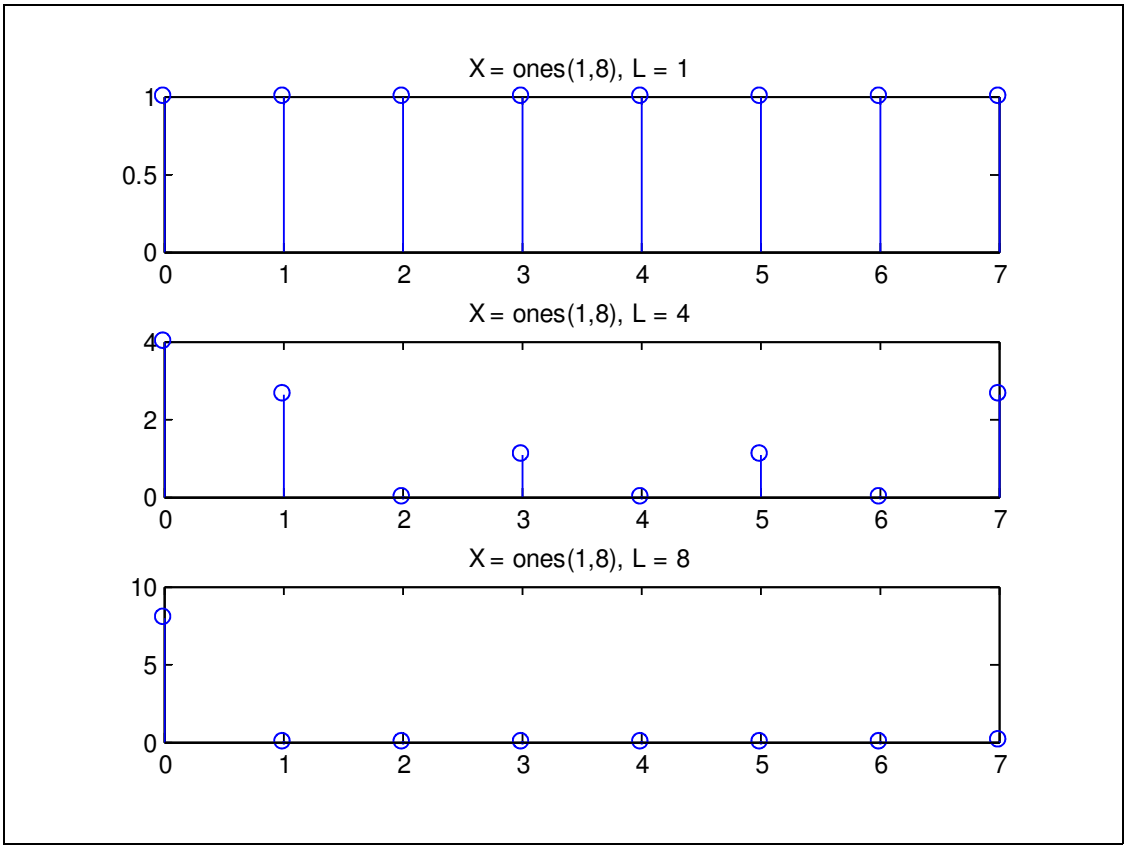
When we do zero padding we are increasing the resolution in the frequency domain meaning that we increase the number of samples of DTFT of the signal.

Problem 2.3

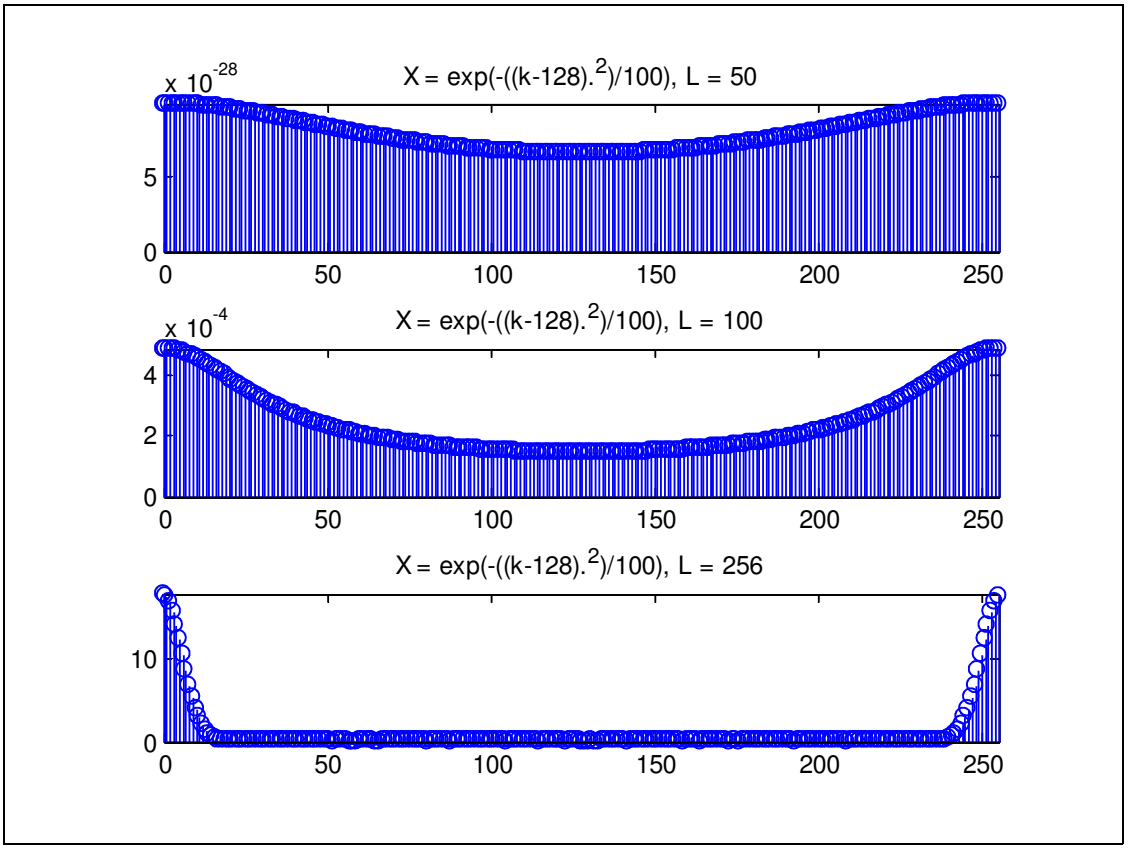
```
function [Y,y] = IDFT(X,L)
N = length(X);
% take less coefficients
Y = X(1:L)
%     inverse dft
y = zeros(1,L);
k = 0:L-1;
   for n = 0:N-1
        y(n+1) = sum(Y.*exp(j*2*pi/N*n*k));
   end
end
```

```
X = ones(1,8);
[H1,h1] = IDFT(X,1);
[H2,h2] = IDFT(X,4);
[H3,h3] = IDFT(X,8);
```

```
figure;
subplot(3,1,1); stem(0:7,abs(h1));title('X = ones(1,8), L = 1');
subplot(3,1,2); stem(0:7,abs(h2));title('X = ones(1,8), L = 4');
subplot(3,1,3); stem(0:7,abs(h3));title('X = ones(1,8), L = 8');
```




```
k = 0:255;
X = exp(-((k-128).^2)/100);
[H1,h1] = IDFT(X,50);
[H2,h2] = IDFT(X,100);
[H3,h3] = IDFT(X,256);

figure;
subplot(3,1,1); stem(0:255,abs(h1));title('X = exp(-((k-128).^2)/100), L =
50');axis tight
subplot(3,1,2); stem(0:255,abs(h2));title('X = exp(-((k-128).^2)/100), L =
100');axis tight
subplot(3,1,3); stem(0:255,abs(h3));title('X = exp(-((k-128).^2)/100), L =
256');axis tight
```



As the number of coefficients increases we see that the signal gets closer to the original one.