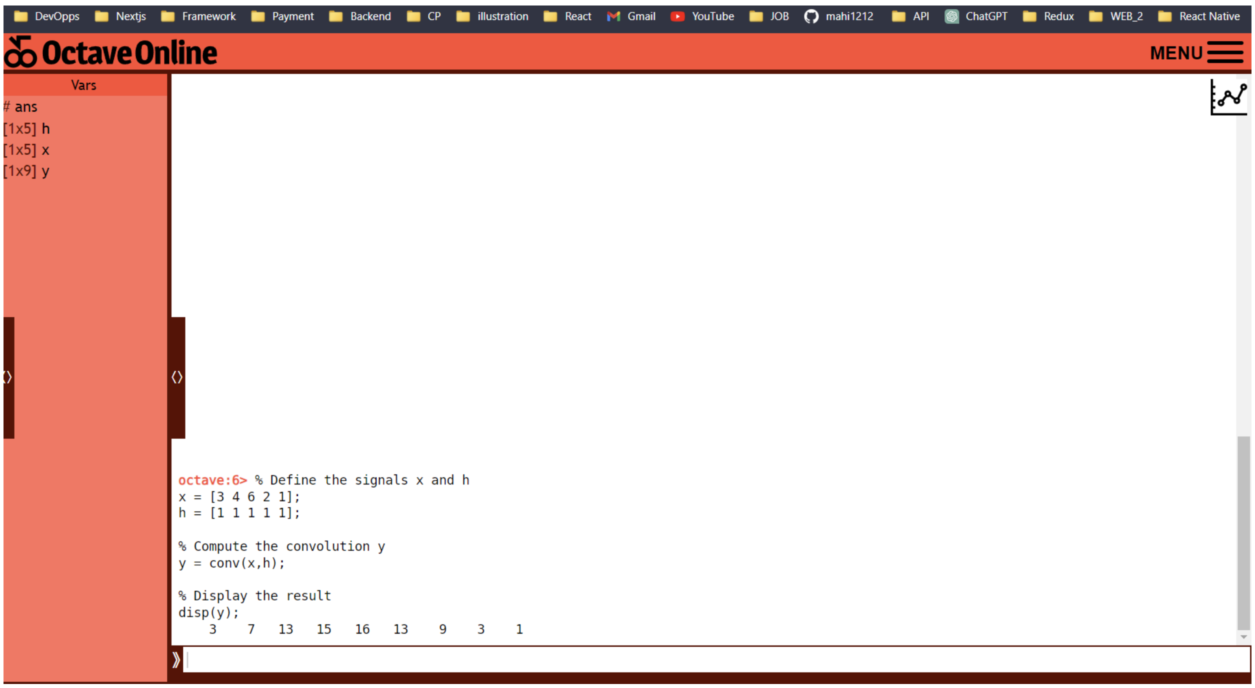
ANS TO THE QUESTION NUMBER: 1

CODE:

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

% Compute the convolution y
y = conv(x,h);
disp(y);
```



ANS TO THE QUESTION NUMBER: 3

Code: % Create a time vector

t = linspace(0, 5, 100);

% Define functions for unit step, unit ramp, unit parabolic, and exponential signals

unit_step = heaviside(t);

unit_ramp = t;

unit_parabolic = t.^2;

exponential = exp(t);

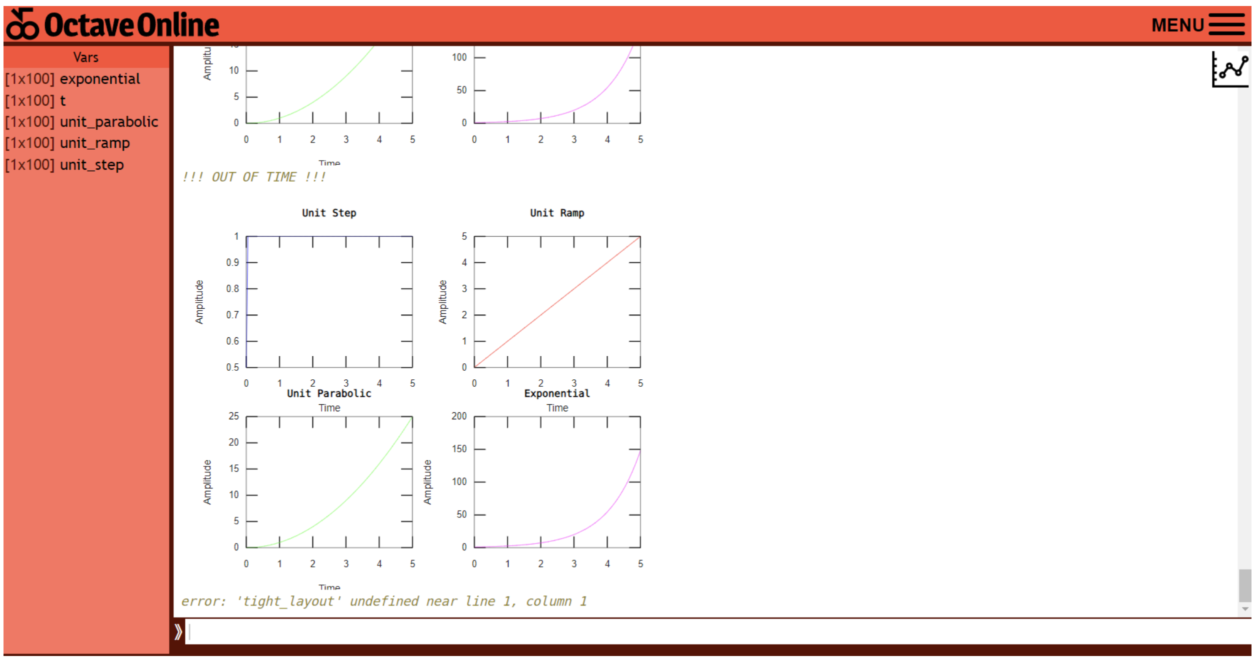
```
% Create a 4x4 subplot grid
subplot(2, 2, 1);
plot(t, unit_step, 'b');
title('Unit Step');
xlabel('Time');
ylabel('Amplitude');

subplot(2, 2, 2);
plot(t, unit_ramp, 'r');
title('Unit Ramp');
xlabel('Time');
ylabel('Amplitude');

subplot(2, 2, 3);
plot(t, unit_parabolic, 'g');
title('Unit Parabolic');
xlabel('Time');
ylabel('Amplitude');

subplot(2, 2, 4);
plot(t, exponential, 'm');
title('Exponential');
xlabel('Time');
```

ylabel('Amplitude');

% Adjust layout for better spacing

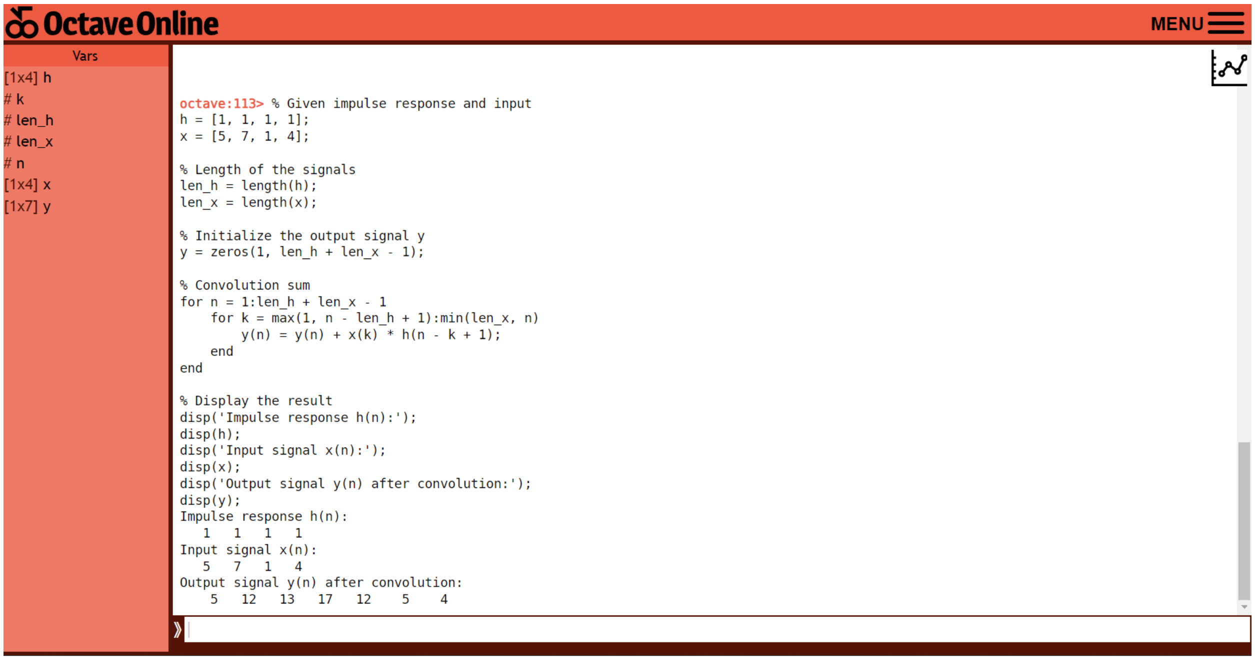
tight_layout();

Output:



ANS TO THE QUESTION NUMBER: 4

Output:

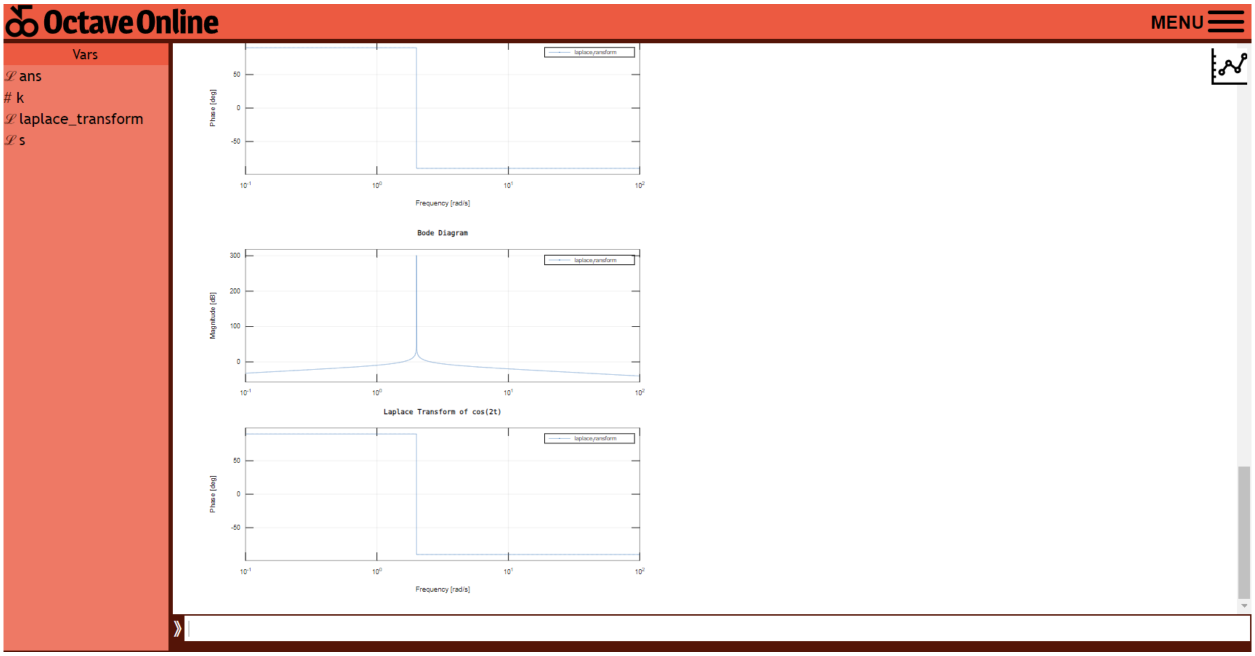


ANS TO THE QUESTION NUMBER: 2

Code:

```
% Define the Laplace variable

s = tf('s');


% Define the frequency of the cosine signal

k = 2;


% Laplace transform of cos(kt)

laplace_transform = s / (s^2 + k^2);


% Plot the Laplace transform

bode(laplace_transform);

title('Laplace Transform of cos(2t)');
```

Output:



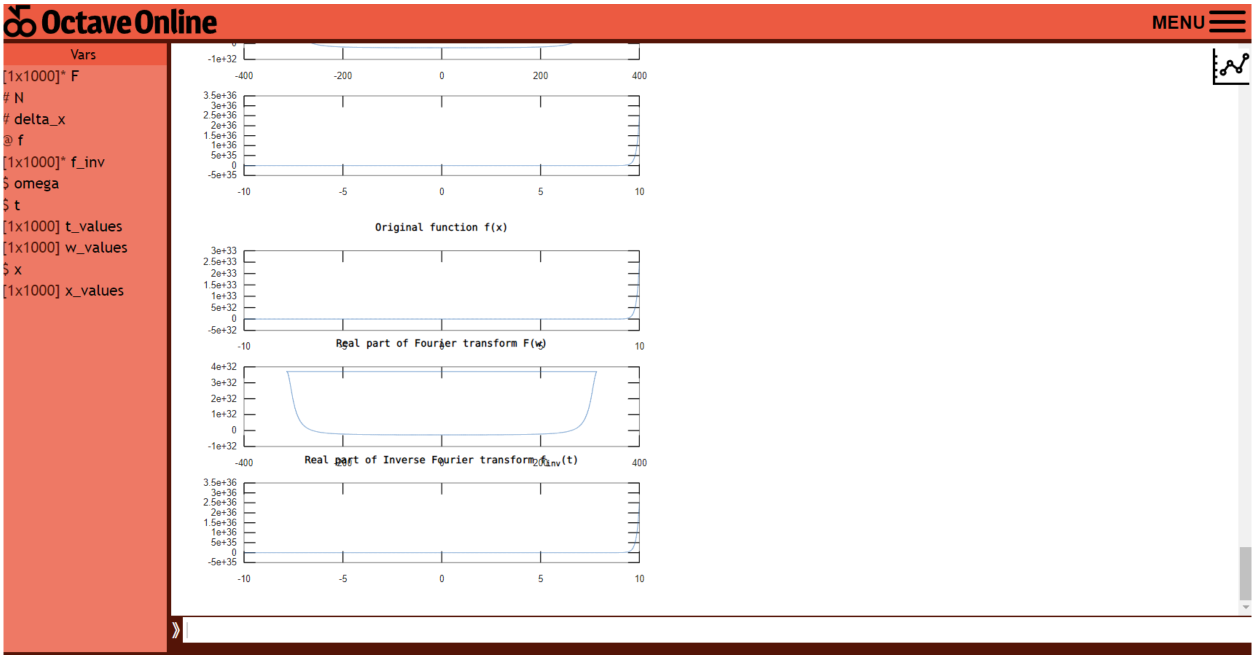
ANS TO THE QUESTION NUMBER: 5

Code:

```
f = @(x) x.^3 .* exp(7*x);

% Define the range and sampling points
x_values = linspace(-10, 10, 1000);
delta_x = x_values(2) - x_values(1);

% Compute Fourier transform using FFT
N = length(x_values);
w_values = fftshift(2 * pi * linspace(-1, 1, N) / delta_x);
F = fftshift(fft(f(x_values), N) * delta_x);
```

```
% Compute inverse Fourier transform using IFFT
t_values = x_values;
f_inv = ifft(fftshift(F) / (2 * pi) * (2 * pi / delta_x), N) * N;

% Display the results
figure;

subplot(3, 1, 1);
plot(x_values, f(x_values));
title('Original function f(x)');

subplot(3, 1, 2);
plot(w_values, real(F));
title('Real part of Fourier transform F(w)');

subplot(3, 1, 3);
plot(t_values, real(f_inv));
title('Real part of Inverse Fourier transform f_{inv}(t)');
```

Output:



ANS TO THE QUESTION NUMBER: 6

Code:

```
% Assume the expression

a = 0.5;

Xz = @(z) 1 / (1 - a * z^(-1));


% Z-transform

z_values = linspace(0, 1, 10);  % Define Z values

Xz_transformed = arrayfun(@(z) Xz(z), z_values);


% Inverse Z-transform (residue method)

n = 0:9;  % Define time indices

xn = a.^n;  % Inverse Z-transform for this simple case
```

```
% Display the results
disp('Original Expression X(z):');
disp(['1 / (1 - ' num2str(a) 'z^(-1))']);
disp('Z-transformed Expression X(z):');
disp(Xz_transformed);
disp('Inverse Z-transformed Expression x(n):');
disp(xn);
```

Output: