MATLAB final project - part 2

Youssef Victor - 6668

```
%% generate_samples helper function
% Generates a samples vector between min and max inputs with the
 specified
% frequency. The length of the output vector is (max - min) *
 frequency.
function f = generate_samples(min, max, frequency)
    if min >= max
        throw(MException('"max" must be greater than "min"'));
    end
    if frequency <= 0
        throw(MException('"frequency" must be positive'));
    end
    f = linspace(min, max, (max - min) * frequency);
end
```

```
%% get_number helper function
% takes a message to display to the user, and keep asking the user to
 enter
% a valid real number based on the validitation predicate.
function f = get_number(prompt, predicate)
    while 1
        n = str2double(input(prompt, 's'));
        if length(n) == 1 && isreal(n) && predicate(n)
            f = n;
            break;
        end
    end
end
```

```
%% get_polynomial_coefficients helper function
% takes a max_power and keep asks the user for the coefficient of a
% polynomial equation with the specified max_power
function f = get_polynomial_coefficients(max_power)
    % For example, third order polynomial has 4 coefficients (a, b, c,
 d).
    % aX^3 + bX^2 + cX^1 + d
    coefficients = zeros(1, max_power + 1);
    for i = 1 : max_power + 1
        coefficients(i) = get_number(['Enter coefficient of X^'
 num2str(max_power - i + 1) ': '], @(x) true);
    end
    f = coefficients;
end
```

```
classdef SignalOperation
   properties(Constant = true)
       NONE = 0
       TIME_SCALING = 1 % Compression / Expansion / Time Reversal
       AMPLITUDE_SCALING = 2
       TIME_SHIFTING = 3
   end

   methods(Static)
       function f = get_operations_message()
           f = [num2str(SignalOperation.NONE) ': None\n'...
                num2str(SignalOperation.TIME_SCALING) ': Time scaling
(compression, expansion, or time reversal)\n'...
                num2str(SignalOperation.AMPLITUDE_SCALING) ':
Amplitude scaling\n'...
                num2str(SignalOperation.TIME_SHIFTING) ': Time
shifting\n'];
       end

       function f = is_valid_signal_operation(signal_operation)
           f = signal_operation == SignalOperation.NONE ||...
               signal_operation == SignalOperation.TIME_SCALING ||...
               signal_operation == SignalOperation.AMPLITUDE_SCALING
||...
               signal_operation == SignalOperation.TIME_SHIFTING;
       end
   end
end



classdef SignalType
   properties(Constant = true)
       DC = 1
       RAMP = 2
       POLYNOMIAL = 3
       EXPONENTIAL = 4
       SINUSOIDAL = 5
   end

   methods(Static)
       function f = get_types_message()
           f = [num2str(SignalType.DC) ': DC\n'...
                num2str(SignalType.RAMP) ': Ramp\n'...
                num2str(SignalType.POLYNOMIAL) ': General order
polynomial\n'...
                num2str(SignalType.EXPONENTIAL) ': Exponential\n'...
                num2str(SignalType.SINUSOIDAL) ': Sinusoidal\n'];
       end

       function f = is_valid_signal_type(signal_type)
```

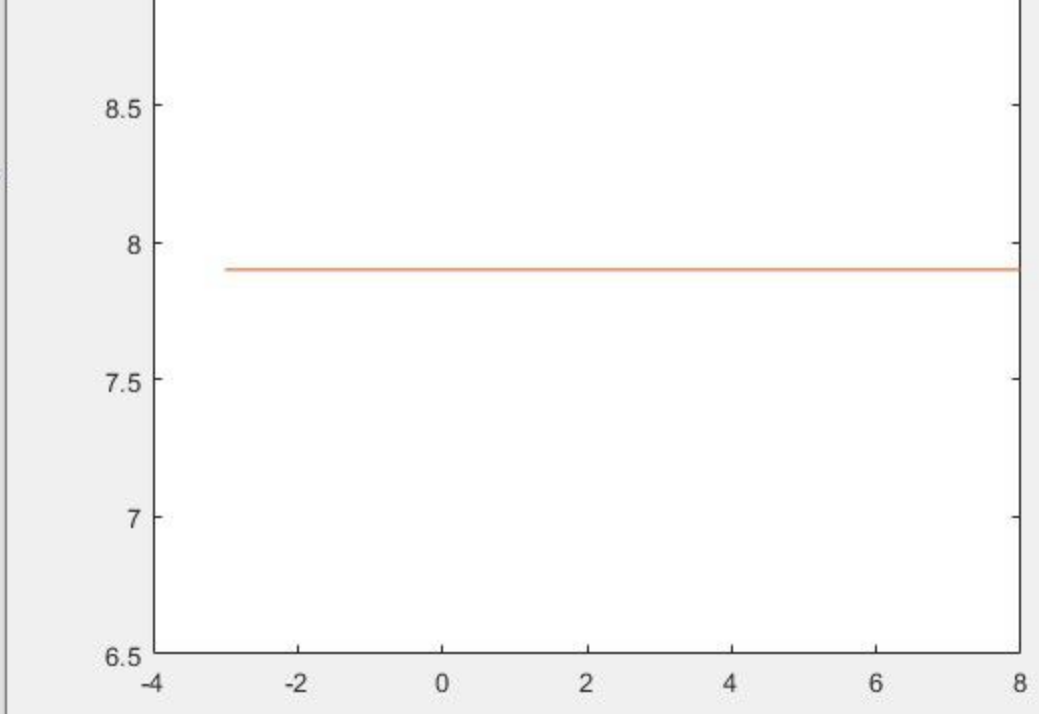
```
            f = signal_type == SignalType.DC ||...
                signal_type == SignalType.RAMP ||...
                signal_type == SignalType.POLYNOMIAL ||...
                signal_type == SignalType.EXPONENTIAL ||...
                signal_type == SignalType.SINUSOIDAL;
        end

        function f = generate(t, predicate)
            f = arrayfun(predicate, t);
        end

        function f = get_dc_function(amplitude)
            f = @(x) amplitude;
        end

        function f = get_ramp_function(slope, intercept)
            f = @(x) slope * x + intercept;
        end

        function f = get_polynomial_function(coefficients)
            f = @(x) polyval(coefficients, x);
        end

        function f = get_exponential_function(amplitude, exponent)
            f = @(x) amplitude * exp(x * exponent);
        end

        function f = get_sinusoidal_function(amplitude, frequency,
 phase)
            f = @(x) amplitude * sin(2 * pi * frequency * x + phase);
        end
    end
 end


sampling_freq = get_number('Enter sampling frequency (must be a
 positive number): ', @(x) x > 0);
t_min = get_number('Enter the start of time scale: ', @(x) true);
t_max = get_number('Enter the end of time scale: ', @(x) x > t_min);

breakpoints_count = get_number('Enter the number of breakpoints (must
 be a non-negative integer): ', @(x) x >= 0 && mod(x, 1) == 0);

breakpoints = zeros(1, breakpoints_count);
for i = 1 : breakpoints_count
    breakpoints(i) = get_number(['Enter the position of breakpoint '
 num2str(i)  ': '],  @(x) x > t_min && x < t_max);
end
breakpoints = [t_min sort(breakpoints) t_max];

t = cell(1, length(breakpoints) - 1);
y = cell(1, length(breakpoints) - 1);

for i = 1 : length(breakpoints) - 1
```

```
    region_min = breakpoints(i);
    region_max = breakpoints(i + 1);
    for_region_string = ['For region ' num2str(region_min) ' to '
 num2str(region_max)];
    % TODO: Drop-down list https://www.mathworks.com/help/matlab/ref/
uidropdown.html
    % TODO: Or, listbox (listdlg)
    signal_type = get_number([for_region_string ', enter a number
 corresponding to the signal type:\n' SignalType.get_types_message()],
 @(x) SignalType.is_valid_signal_type(x));
    current_t = generate_samples(region_min, region_max,
 sampling_freq);
    t(i)= {current_t};
    switch signal_type
        case SignalType.DC
            amplitude = get_number('Enter amplitude for DC signal: ',
 @(x) true);
            y(i) = {SignalType.generate(current_t,
 SignalType.get_dc_function(amplitude))};
        case SignalType.RAMP
            slope = get_number('Enter slope for ramp signal: ', @(x)
 true);
            intercept = get_number('Enter intercept for ramp signal:
 ', @(x) true);
            y(i) = {SignalType.generate(current_t,
 SignalType.get_ramp_function(slope, intercept))};
        case SignalType.POLYNOMIAL
            max_power = get_number('Enter the order (max power) of the
 polynomial: ', @(x) x > 0);
            coefficients = get_polynomial_coefficients(max_power);
            y(i) = {SignalType.generate(current_t,
 SignalType.get_polynomial_function(coefficients))};
        case SignalType.EXPONENTIAL
            amplitude = get_number('Enter amplitude for exponential
 signal: ', @(x) true);
            exponent = get_number('Enter exponent for exponential
 signal: ', @(x) true);
            y(i) = {SignalType.generate(current_t,
 SignalType.get_exponential_function(amplitude, exponent))};
        case SignalType.SINUSOIDAL
            amplitude = get_number('Enter amplitude for sinusoidal
 signal: ', @(x) true);
            frequency = get_number('Enter frequency for sinusoidal
 signal: ', @(x) true);
            phase = get_number('Enter phase for sinusoidal signal: ',
 @(x) true);
            y(i) = {SignalType.generate(current_t,
 SignalType.get_sinusoidal_function(amplitude, frequency, phase))};
    end
end

tVector = [t{:}];
yVector = [y{:}];
plot(tVector, yVector);
```

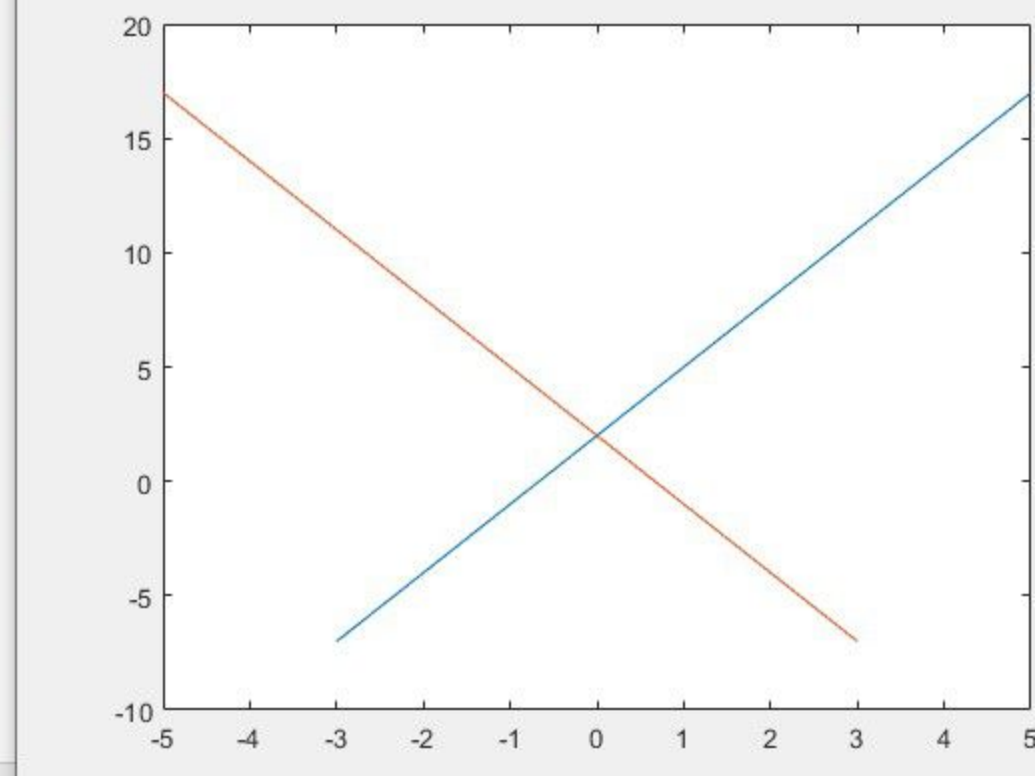
```
datacursormode on;

signal_operation = get_number(['Enter a number
 corresponding to the signal operation you want:\n'
 SignalOperation.get_operations_message()], @(x)
 SignalOperation.is_valid_signal_operation(x));
switch signal_operation
    case SignalOperation.TIME_SCALING
        scaling_factor = get_number('Enter scaling factor: ', @(x)
 true);
        tVector = tVector * scaling_factor;
    case SignalOperation.AMPLITUDE_SCALING
        scaling_factor = get_number('Enter scaling factor: ', @(x)
 true);
        yVector = yVector * scaling_factor;
    case SignalOperation.TIME_SHIFTING
        shifting_value = get_number('Enter shifting value (positive
 shifts to left, negative to right): ', @(x) true);
        tVector = tVector - shifting_value;
end

hold on;
plot(tVector, yVector);
datacursormode on;
```

Published with MATLAB® R2016b

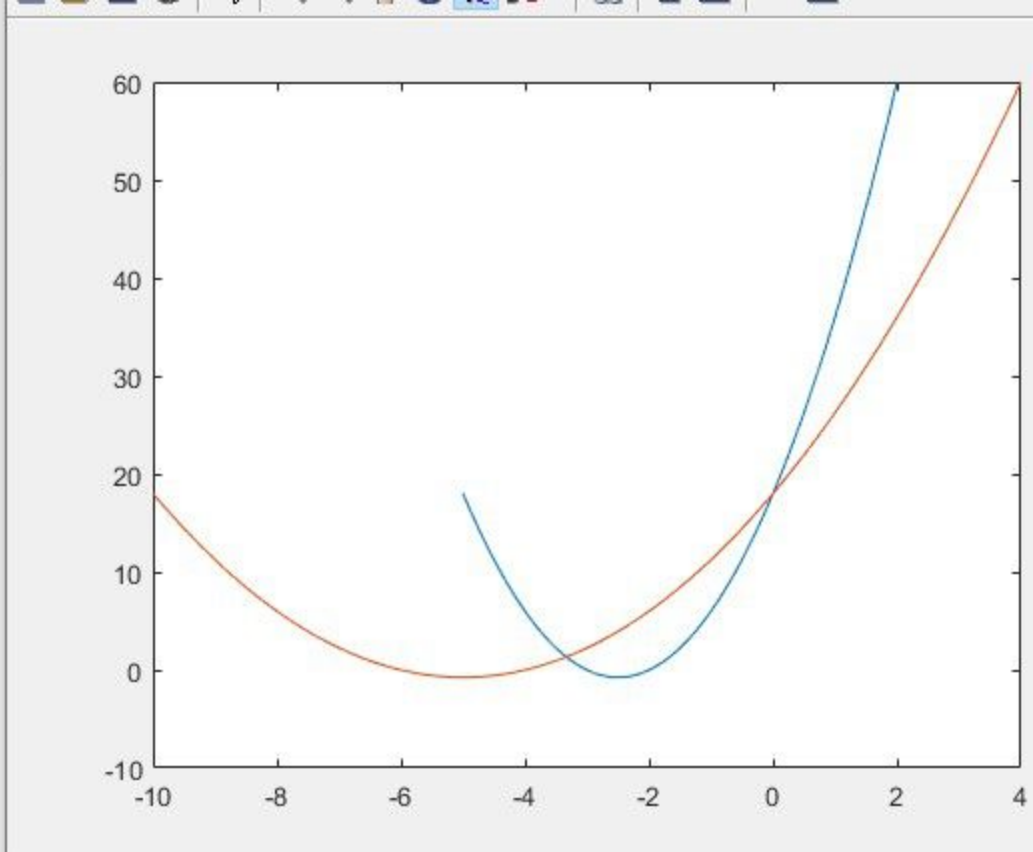
```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -3
Enter the end of time scale: 8
Enter the number of breakpoints (must be a non-negative integer): 0
For region -3 to 8, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 7.9
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
0
>>
```



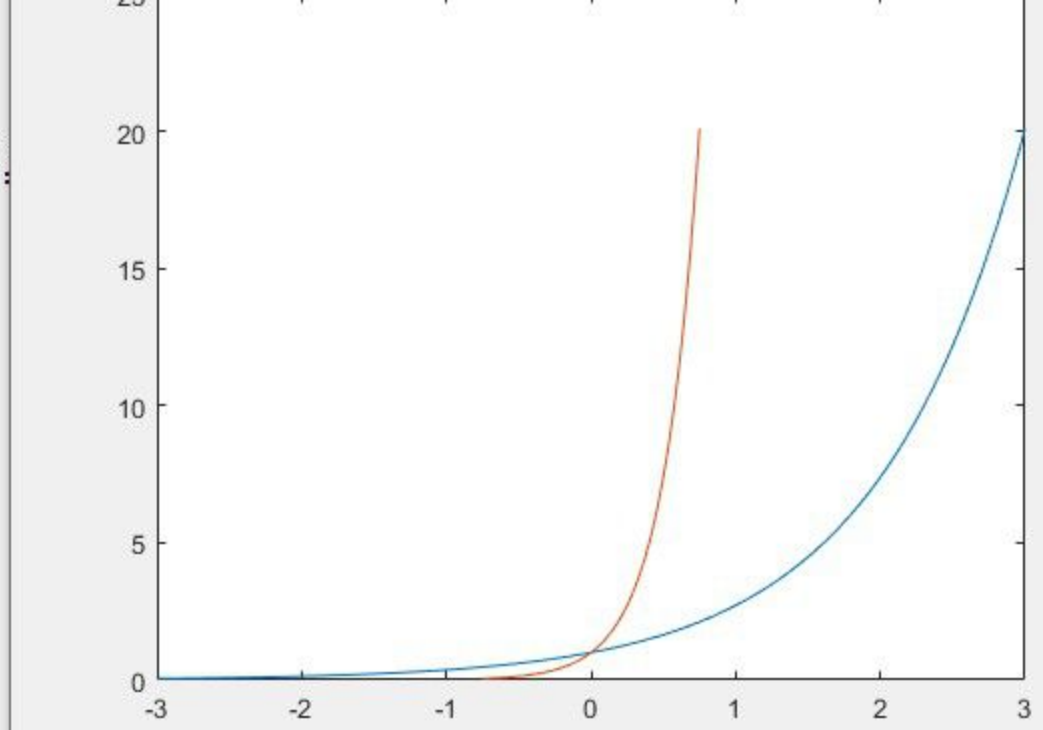
```
Enter sampling frequency (must be a positive number): 200
Enter the start of time scale: -3
Enter the end of time scale: 5
Enter the number of breakpoints (must be a non-negative integer): 0
For region -3 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
2
Enter slope for ramp signal: 3
Enter intercept for ramp signal: 2
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
1
Enter scaling factor: -1
>>
```



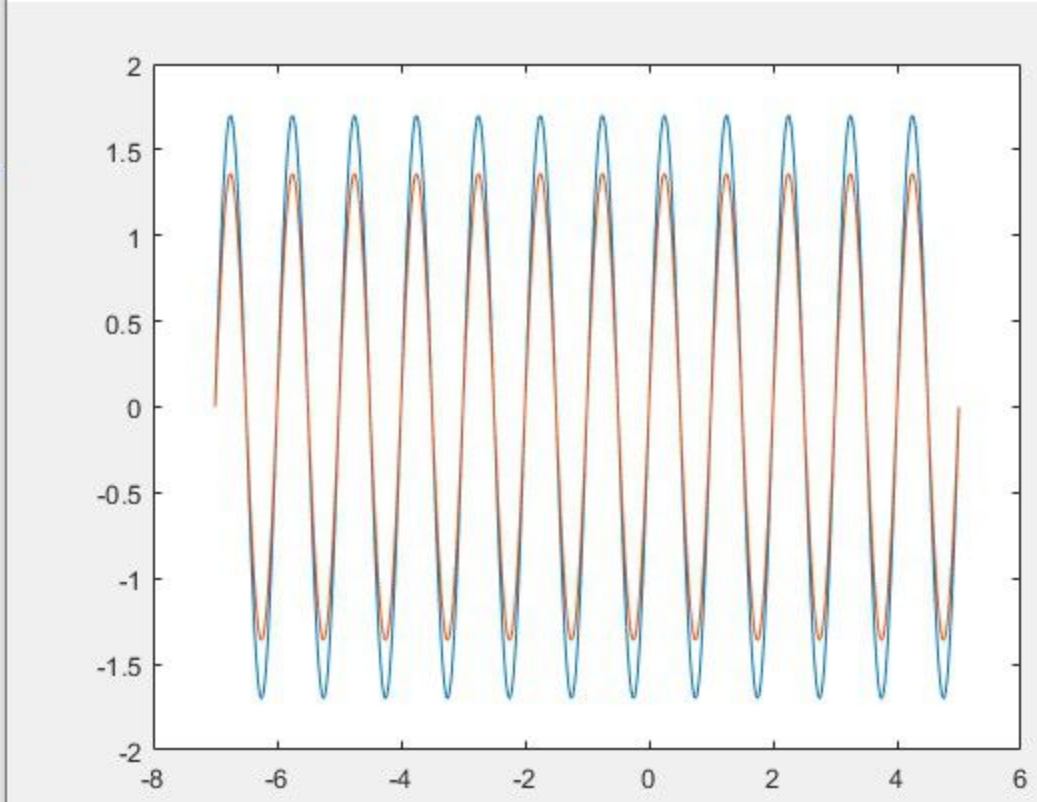
```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -5
Enter the end of time scale: 2
Enter the number of breakpoints (must be a non-negative integer): 0
For region -5 to 2, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
3
Enter the order (max power) of the polynomial: 2
Enter coefficient of X^2: 3
Enter coefficient of X^1: 15
Enter coefficient of X^0: 18
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
1
Enter scaling factor: 2
>>
```



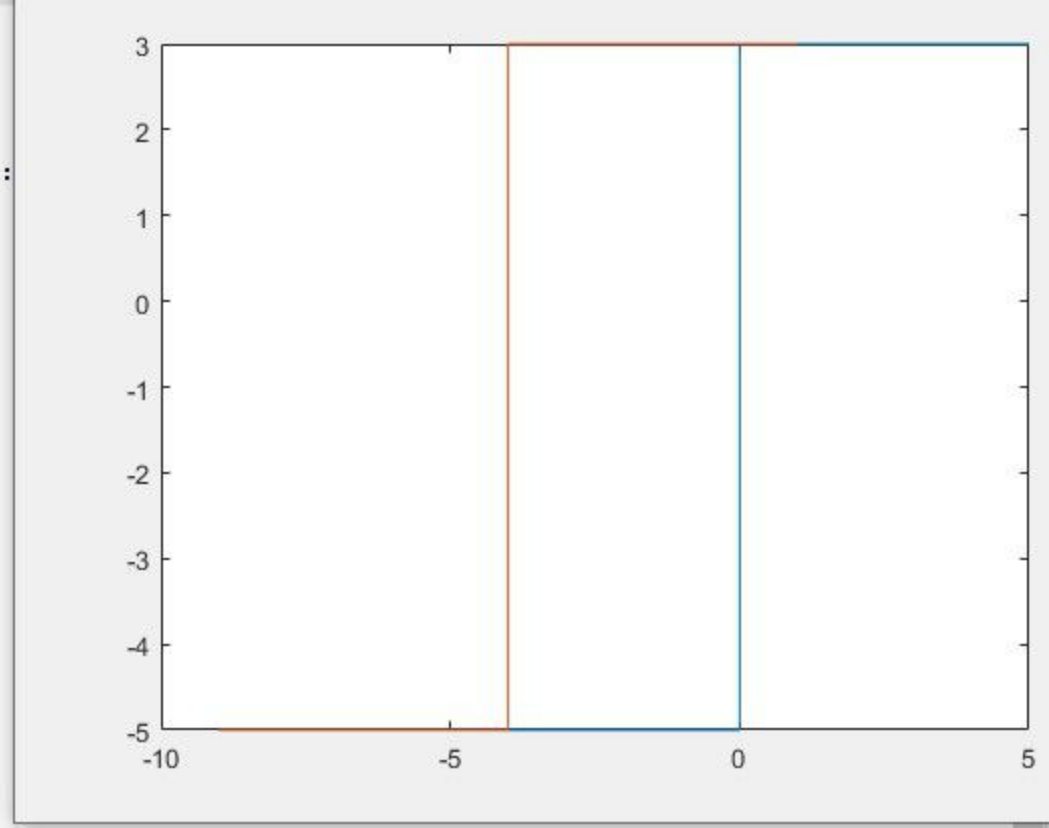
```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -3
Enter the end of time scale: 3
Enter the number of breakpoints (must be a non-negative integer): 0
For region -3 to 3, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
4
Enter amplitude for exponential signal: 1
Enter exponent for exponential signal: 1
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
1
Enter scaling factor: 0.25
```



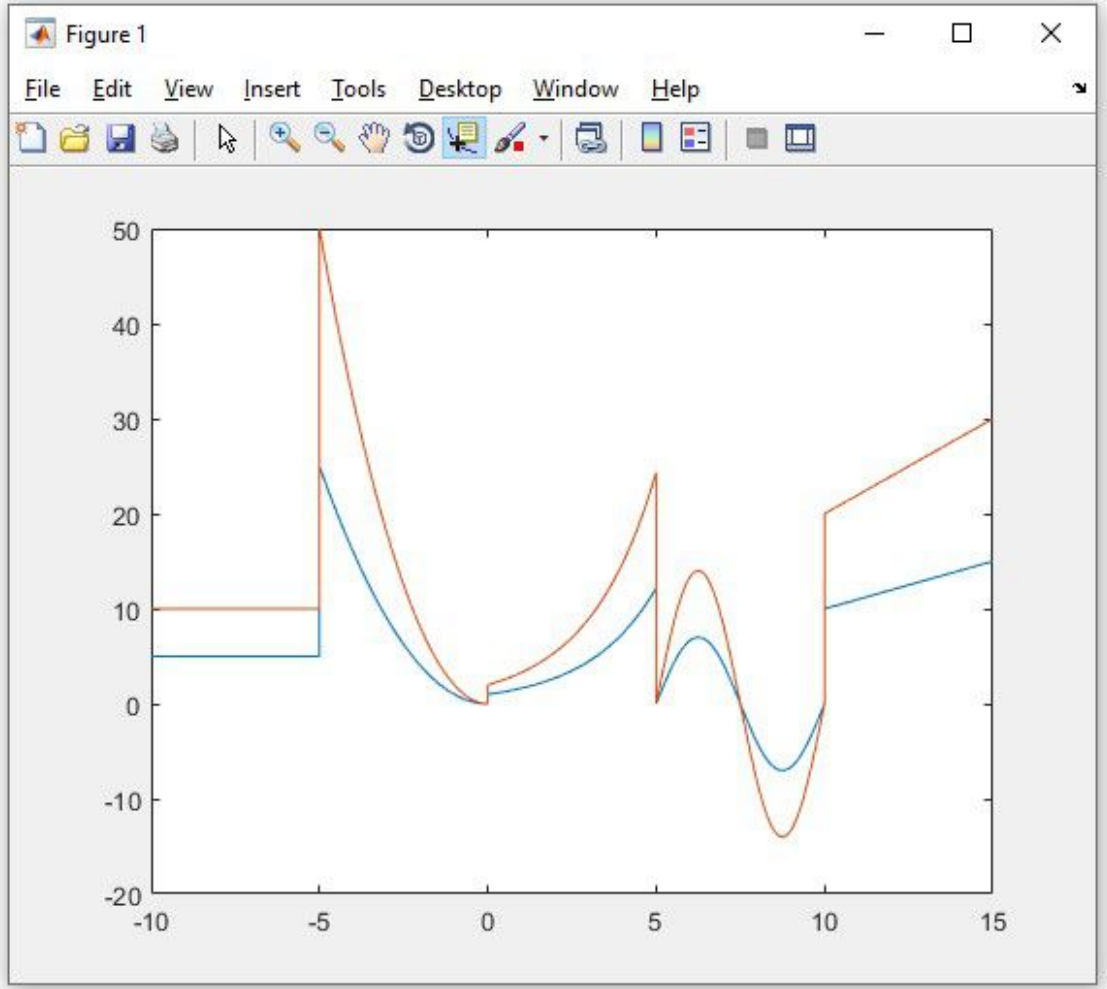
```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -7
Enter the end of time scale: 5
Enter the number of breakpoints (must be a non-negative integer): 0
For region -7 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
5
Enter amplitude for sinusoidal signal: 1.7
Enter frequency for sinusoidal signal: 1
Enter phase for sinusoidal signal: 0
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
2
Enter scaling factor: 0.8
>>
```



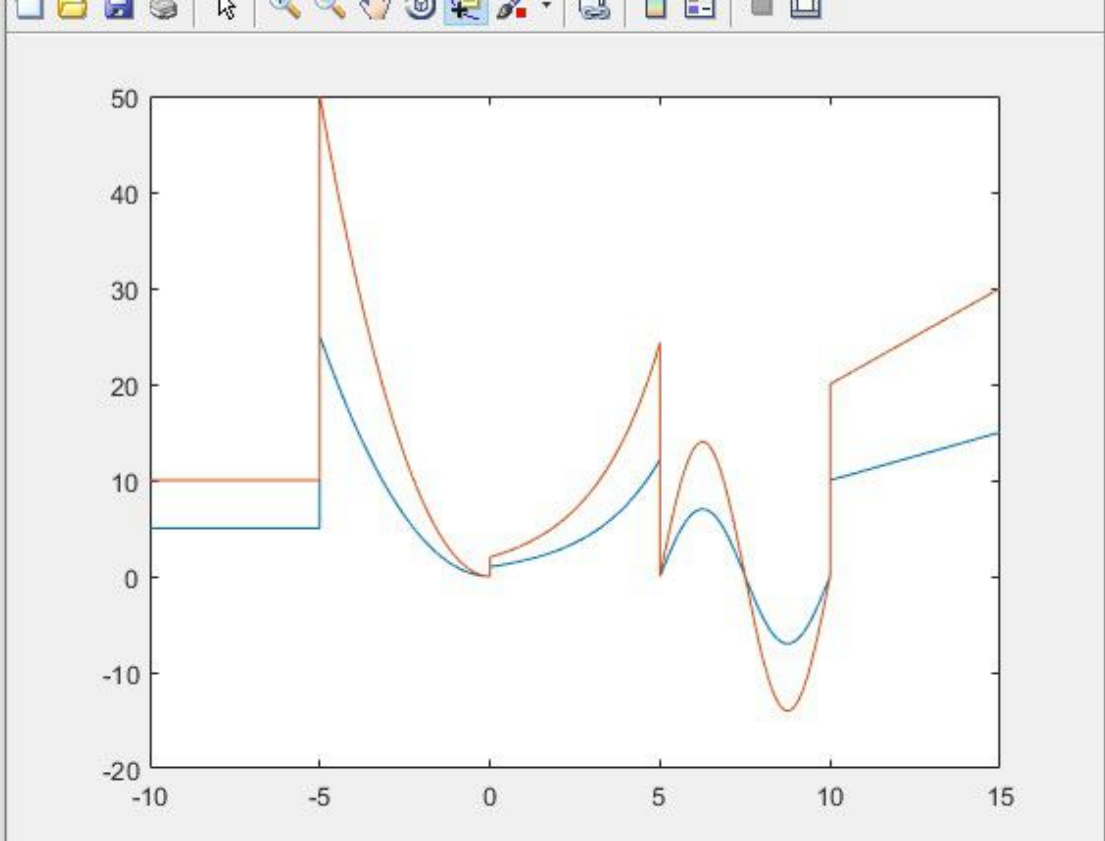
```
Enter the start of time scale: -5
Enter the end of time scale: 5
Enter the number of breakpoints (must be a non-negative integer): 1
Enter the position of breakpoint 1: 0
For region -5 to 0, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: -5
For region 0 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 3
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
3
Enter shifting value (positive shifts to left, negative to right): 4
```



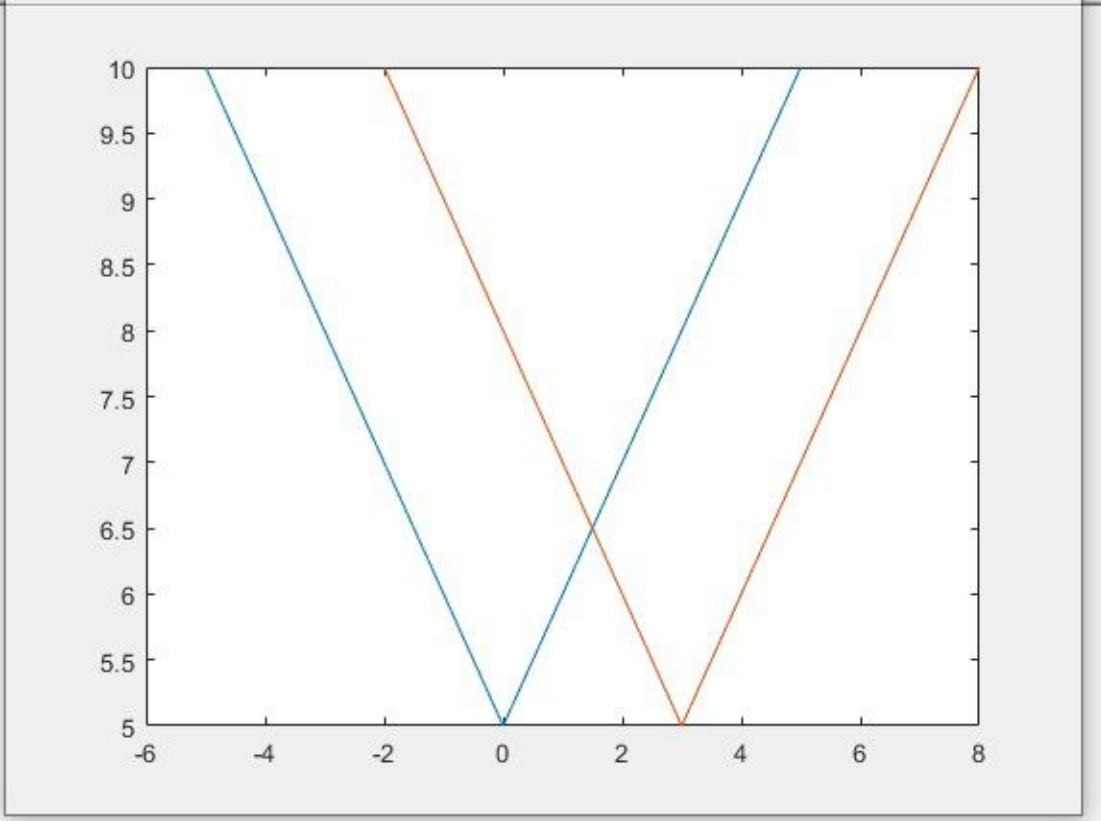
```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -10
Enter the end of time scale: 15
Enter the number of breakpoints (must be a non-negative integer): 4
Enter the position of breakpoint 1: -5
Enter the position of breakpoint 2: 0
Enter the position of breakpoint 3: 5
Enter the position of breakpoint 4: 10
For region -10 to -5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 5
For region -5 to 0, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
3
Enter the order (max power) of the polynomial: 2
Enter coefficient of X^2: 1
Enter coefficient of X^1: 0
Enter coefficient of X^0: 0
For region 0 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
4
Enter amplitude for exponential signal: 1
Enter exponent for exponential signal: 0.5
```



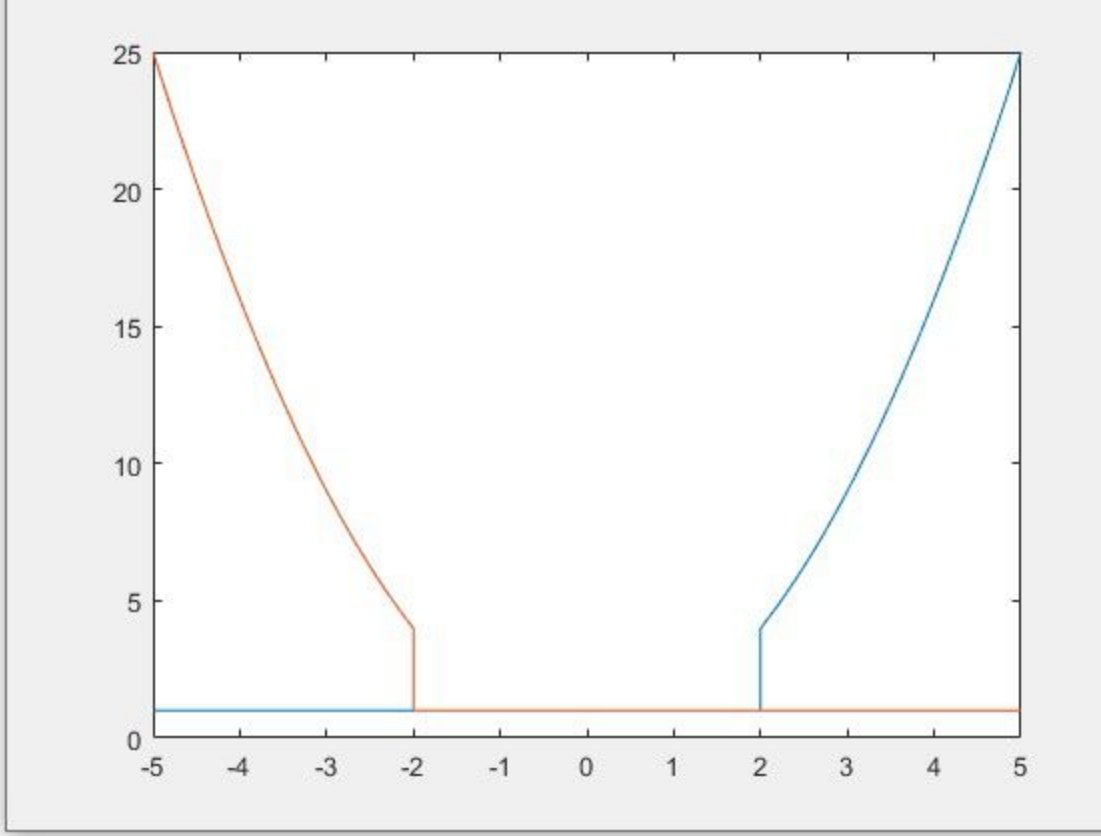
```
For region 5 to 10, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
5
Enter amplitude for sinusoidal signal: 7
Enter frequency for sinusoidal signal: 0.2
Enter phase for sinusoidal signal: 0
For region 10 to 15, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
2
Enter slope for ramp signal: 1
Enter intercept for ramp signal: 0
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
2
Enter scaling factor: 2
```



```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -5
Enter the end of time scale: 5
Enter the number of breakpoints (must be a non-negative integer): 1
Enter the position of breakpoint 1: 0
For region -5 to 0, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
2
Enter slope for ramp signal: -1
Enter intercept for ramp signal: 5
For region 0 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
2
Enter slope for ramp signal: 1
Enter intercept for ramp signal: 5
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
3
Enter shifting value (positive shifts to left, negative to right): -3
```



```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -5
Enter the end of time scale: 5
Enter the number of breakpoints (must be a non-negative integer): 1
Enter the position of breakpoint 1: 2
For region -5 to 2, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 1
For region 2 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
3
Enter the order (max power) of the polynomial: 2
Enter coefficient of X^2: 1
Enter coefficient of X^1: 0
Enter coefficient of X^0: 0
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
1
Enter scaling factor: -1
```



```
Enter sampling frequency (must be a positive number): 100
Enter the start of time scale: -7
Enter the end of time scale: 5
Enter the number of breakpoints (must be a non-negative integer): 2
Enter the position of breakpoint 1: -3
Enter the position of breakpoint 2: -1
For region -7 to -3, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 6
For region -3 to -1, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 4.5
For region -1 to 5, enter a number corresponding to the signal type:
1: DC
2: Ramp
3: General order polynomial
4: Exponential
5: Sinusoidal
1
Enter amplitude for DC signal: 8.2
Enter a number corresponding to the signal operation you want:
0: None
1: Time scaling (compression, expansion, or time reversal)
2: Amplitude scaling
3: Time shifting
1
Enter scaling factor: -1
```