
```
% ENGR 451 - Chapter 1 Laboratory
% Matlab tutorial

clear
x.data = [1 2 3 4 5];
x.offset = 1;
y.data = [5 4 3];
y.offset = 0;

% test add
test_lab1_2024('addit(x, y)')
test_lab1_2024('addit(y, x)')
test_lab1_2024('addit(1, x)')
test_lab1_2024('addit(x, 1)')

y.offset = 2;
test_lab1_2024('addit(x, y)')
test_lab1_2024('addit(y, x)')

y.offset = 7;
% test sub
test_lab1_2024('subit(x, y)')
test_lab1_2024('subit(y, x)')
test_lab1_2024('subit(1, x)')
test_lab1_2024('subit(x, 1)')

% test mult
test_lab1_2024('multit(x, y)')
test_lab1_2024('multit(3, x)')
test_lab1_2024('multit(x, 3)')

% test flip
test_lab1_2024('flipit(x)')

% test shift
test_lab1_2024('shiftit(y, 2)')

% test trim
x.data = [0 0 1 2 3 0];
test_lab1_2024('trimit(x)')

%combinations
test_lab1_2024('flipit(subit(shiftit(addit(x, 2), 4), y))')
test_lab1_2024('addit(flipit(addit(x, y)), shiftit(y, -5))')
test_lab1_2024('subit(addit(multit(shiftit(flipit(x), 4), shiftit(y, 3)),
flipit(y)), x)')

% test stem
clf
stem(y)
grid on
ax = axis;
```

```

set(gca, 'XTick', ax(1):ax(2), ...
    'YTick', ax(3):ax(4))
ch = get(gca, 'Child');
ch.MarkerFaceColor = 'b';

% Program Listings
fprintf('\n\n')
disp('--- flipit.m -----')
type flipit
disp(' ')
disp('--- shiftit.m -----')
type shiftit
disp(' ')
disp('--- addit.m -----')
type addit
disp(' ')
disp('--- subit.m -----')
type subit
disp(' ')
disp('--- multit.m -----')
type multit
disp(' ')
disp('--- trimit.m -----')
type trimit
disp(' ')
disp('--- stemit.m -----')
type stemit

addit(x, y): sequence O.K.
addit(y, x): sequence O.K.
addit(1, x): sequence O.K.
addit(x, 1): sequence O.K.
addit(x, y): sequence O.K.
addit(y, x): sequence O.K.
subit(x, y): sequence O.K.
subit(y, x): sequence O.K.
subit(1, x): sequence O.K.
subit(x, 1): sequence O.K.
multit(x, y): sequence O.K.
multit(3, x): sequence O.K.
multit(x, 3): sequence O.K.
flipit(x): sequence O.K.
shiftit(y, 2): sequence O.K.
trimit(x): sequence O.K.
flipit(subit(shiftit(addit(x, 2), 4), y)): sequence O.K.
addit(flipit(addit(x, y)), shiftit(y, -5)): sequence O.K.
subit(addit(multit(shiftit(flipit(x), 4), shiftit(y, 3)), flipit(y)), x):
sequence O.K.

ans =

    Stem with properties:

        Color: [0 0.4470 0.7410]

```

```
LineStyle: '-'
LineWidth: 0.5000
Marker: 'o'
MarkerSize: 6
MarkerFaceColor: 'none'
BaseValue: 0
XData: [7 8 9]
YData: [5 4 3]
```

Use GET to show all properties

```
--- flipit.m -----
```

```
function y = flipit(x)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
x_len = length(x.data);
x_index_right = x.offset + (x_len-1);

y.data = x.data(end:-1:1);
y.offset = -x_index_right;
end
```

```
--- shiftit.m -----
```

```
function y = shiftit(x, n0)
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes here
y.data = x.data;
y.offset = x.offset + n0;
end
```

```
--- addit.m -----
```

```
function z = addit(x,y)

if ~isstruct(x)
    z.data = y.data + x;
    z.offset = y.offset;
elseif ~isstruct(y)
    z.data = x.data + y;
    z.offset = x.offset;
else

    %construct z vector
    min_index = min(x.offset, y.offset);
    max_index = max(x.offset + (length(x.data)-1), y.offset +
(length(y.data)-1));
    z_length = max_index-min_index+1;
    z.data = zeros(1,z_length);

    x_len = length(x.data);
```

```

    y_len = length(y.data);

    %calculate offsets in z
    x_start_in_z = x.offset-min_index+1;
    y_start_in_z = y.offset-min_index+1;

    x_end_in_z = x_start_in_z + x_len - 1;
    y_end_in_z = y_start_in_z + y_len - 1;

    z.data(x_start_in_z:x_end_in_z) = z.data(x_start_in_z:x_end_in_z) +
x.data;
    z.data(y_start_in_z:y_end_in_z) = z.data(y_start_in_z:y_end_in_z) +
y.data;

    z.offset = min(x.offset, y.offset);
end

end

--- subit.m -----

function z = subit(x,y)

    if ~isstruct(x)
        z.data = x - y.data;
        z.offset = y.offset;
    elseif ~isstruct(y)
        z.data = x.data - y;
        z.offset = x.offset;
    else

        %construct z vector
        min_index = min(x.offset, y.offset);
        max_index = max(x.offset + (length(x.data)-1), y.offset +
(length(y.data)-1));
        z_length = max_index-min_index+1;
        z.data = zeros(1,z_length);

        x_len = length(x.data);
        y_len = length(y.data);

        %calculate offsets in z
        x_start_in_z = x.offset-min_index+1;
        y_start_in_z = y.offset-min_index+1;

        x_end_in_z = x_start_in_z + x_len - 1;
        y_end_in_z = y_start_in_z + y_len - 1;

        z.data(x_start_in_z:x_end_in_z) = x.data;
        z.data(y_start_in_z:y_end_in_z) = z.data(y_start_in_z:y_end_in_z) -
y.data;

```

```

        z.offset = min(x.offset, y.offset);
    end
end

--- multit.m -----

function z=multit(x,y)

    if ~isstruct(x)
        z.data = y.data * x;
        z.offset = y.offset;
    elseif ~isstruct(y)
        z.data = x.data * y;
        z.offset = x.offset;
    else

        x_len = length(x.data);
        y_len = length(y.data);

        x_index_left = x.offset;
        x_index_right = x.offset + (x_len - 1);

        y_index_left = y.offset;
        y_index_right = y.offset + (y_len - 1);

        %construct z vector
        min_index = min(x_index_left, y_index_left);
        max_index = max(x_index_right, y_index_right);
        z_length = max_index-min_index+1;
        z.data = ones(1,z_length);
        %ok

        %pad x
        left_size = x_index_left-min_index;
        right_size = max_index-x_index_right;
        left_pad = zeros(1, left_size);
        right_pad = zeros(1, right_size);
        x.data = [left_pad, x.data, right_pad];

        %pad y
        left_size = y_index_left-min_index;
        right_size = max_index-y_index_right;
        left_pad = zeros(1, left_size);
        right_pad = zeros(1, right_size);
        y.data = [left_pad, y.data, right_pad];

        z.data = x.data .* y.data;
        z.offset = min(x.offset, y.offset);
    end

end

--- trimit.m -----

```

```

function z = trimit(x)

    find_i_left = find(x.data, 1, 'first');
    find_i_right = find(x.data, 1, 'last');

    z.offset = find_i_left + x.offset - 1;
    z.data = x.data(find_i_left:find_i_right);
end

```

```

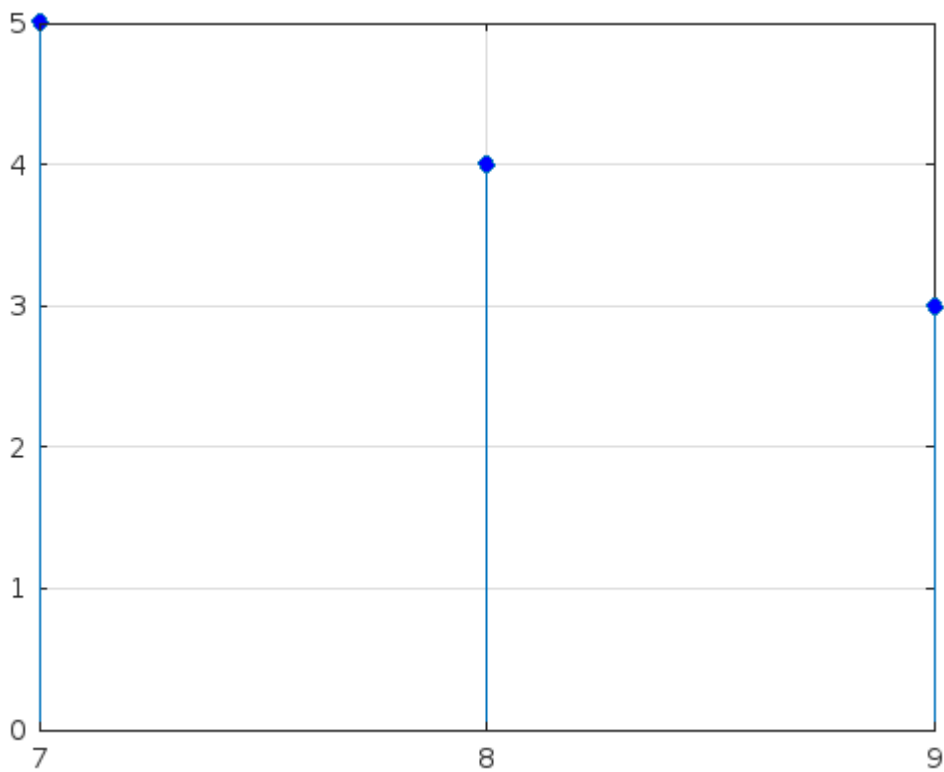
--- stemit.m -----

```

```

function z = stemit(x)
    n = length(x.data);
    x_vector = (x.offset:((n-1) + x.offset));
    z = stem(x_vector, x.data);
end

```



Published with MATLAB® R2023b