Table of Contents

Problem 1

```
clear
clc

disp("*********************" + newline + "Problem 1" + newline);

%
 ************************************************************************
% Part a
disp("Part a" + newline);

% Calculates 100 regularly spaced intervals then creates a vector and
% prints length to console
y = (28-5) / (100 - 1);
x = 5:y:28;
disp("Method one Length of x = " + length(x));

% Automatically creates a vector of 100 regularly spaced intervals and
% prints length to console
x = linspace(5, 28, 100);
disp("Method two Length of x = " + length(x));

%
 ************************************************************************
% Part b
disp(newline + "Part b" + newline);

% Calculates the regularly spacing needed to create a vector of
 spacing 0.2
% and prints length to console
y = ((14 - 2) * (1 / .2)) + 1;
x = linspace(2, 14, y);
disp("Method one Length of x = " + length(x));

% Calculates the regularly spacing needed to create a vector of
 spacing 0.2
% and prints length to console
```

```
 x = 2:0.2:14;
disp("Method two Length of x = " + length(x));

%
 ************************************************************************
% Part c
disp(newline + "Part c" + newline);

% Calculates 50 regularly spaced intervals then creates a vector and
% prints length to console
y = (5 + 2) / (50 - 1);
x = -2:y:5;
disp("Method one Length of x = " + length(x));

% Automatically creates a vector of 50 regularly spaced intervals and
% prints length to console
x = linspace(-2, 5, 50);
disp("Method two Length of x = " + length(x));
```

```
*********************
Problem 1

Part a

Method one Length of x = 100
Method two Length of x = 100

Part b

Method one Length of x = 61
Method two Length of x = 61

Part c

Method one Length of x = 50
Method two Length of x = 50
```

Problem 2

```
clear

disp(newline + "*********************" + newline + "Problem 2" +
 newline);

%
 ************************************************************************
% Part a
disp("Part a" + newline);

% Calculates 50 regularly spaced values using the base 10 power of the
% starting and ending numbers
x = logspace(1, 3, 50);
disp("Length of x = " + length(x));
```

```
%
 ************************************************************************
% Part b
disp(newline + "Part b" + newline);

% Calculates 20 regularly spaced values using the base 10 power of the
% starting and ending numbers
x = logspace(1, 3, 20);
disp("Length of x = " + length(x));
```

```
********************
Problem 2

Part a

Length of x = 50

Part b

Length of x = 20
```

Problem 3

```
clear

disp(newline + "********************" + newline + "Problem 3" +
 newline);

% Create vector x
x = linspace(0, 10, 6);

% Create array A
A = [3 * x; (5 * x) - 20];

% Display array A to the console
disp(A);
```

```
********************
Problem 3

     0     6    12    18    24    30
   -20   -10     0    10    20    30
```

Problem 4

```
clear

disp(newline + "********************" + newline + "Problem 4" +
 newline);
```

```
% Create vector x
x = linspace(0, 10, 6);

% Create array A
A = [3 * x' ((5 * x) - 20)'];

% Display array A to the console
disp(A);


*********************
Problem 4

     0   -20
     6   -10
    12     0
    18    10
    24    20
    30    30
```

Problem 5

```
clear

disp(newline + "*********************" + newline + "Problem 5" +
 newline);

% Create matrix A
A = [3, 7, -4, 12;
    -5, 9, 10, 2;
    6, 13, 8, 11;
    15, 5, 4, 1];

% Create vector v and print to console
v = A(:, 2);
disp("v =");
disp(v);

% Create vector w and print to console
w = A(2, :);
disp("w =");
disp(w);


*********************
Problem 5

v =
     7
     9
    13
```

```
     5

w =
    -5     9    10     2
```

Problem 6

```
clear

disp(newline + "*********************" + newline + "Problem 6" +
 newline);

% Create matrix A
A = [3, 7, -4, 12;
    -5, 9, 10, 2;
    6, 13, 8, 11;
    15, 5, 4, 1];

% Create 4x3 array B and print to console
B = A(:, 2:4);
disp("B =");
disp(B);

% Create 3x4 array C and print to console
C = A(2:4, :);
disp("C =");
disp(C);

% Create 2x3 array D and print to console
D = A(1:2, 2:4);
disp("D =");
disp(D);
```

```

*********************
Problem 6

B =
     7    -4    12
     9    10     2
    13     8    11
     5     4     1

C =
    -5     9    10     2
     6    13     8    11
    15     5     4     1

D =
     7    -4    12
     9    10     2
```

Problem 7

```
clear

disp(newline + "*********************" + newline + "Problem 7" +
 newline);

% Create known vectors
x = [2, 4, 7];
y = [2, -4, 7];
z = [5+3i, -3+4i, 2-7i];

% Length and absolute value of x
disp("Length of x = " + length(x));
disp("Absolute value of x = ");
disp(abs(x));

% Length and absolute value of y
disp("Length of y = " + length(y));
disp("Absolute value of y = ");
disp(abs(y));

% Length and absolute value of z
disp("Length of z = " + length(z));
disp("Absolute value of z = ");
disp(abs(z));
```

```
*********************
Problem 7

Length of x = 3
Absolute value of x = 
     2     4     7

Length of y = 3
Absolute value of y = 
     2     4     7

Length of z = 3
Absolute value of z = 
    5.8310    5.0000    7.2801
```

Problem 8

```
clear

disp(newline + "*********************" + newline + "Problem 8" +
 newline);

% Create matrix A
```

```
A = [3, 7, -4, 12;
    -5, 9, 10, 2;
    6, 13, 8, 11;
    15, 5, 4, 1];

%
 ************************************************************************
% Part a
disp("Part a" + newline);

% Find min and max value in each column of A
maxCA = max(A);
minCA = min(A);
disp("Maximum in each column of A = ");
disp(maxCA);
disp("Minimum in each column of A = ");
disp(minCA);

%
 ************************************************************************
% Part b
disp("Part b" + newline);

% Find min and max value in each row of A
maxRA = max(A'); %#ok<*UDIM>
minRA = min(A');
disp("Maximum in each row of A = ");
disp(maxRA);
disp("Minimum in each row of A = ");
disp(minRA);
```

```
*********************
Problem 8

Part a

Maximum in each column of A =
    15    13    10    12

Minimum in each column of A =
    -5     5    -4     1

Part b

Maximum in each row of A =
    12    10    13    15

Minimum in each row of A =
    -4    -5     6     1
```

Problem 9

```
clear

disp(newline + "*********************" + newline + "Problem 9" +
 newline);

% Create matrix A
A = [3, 7, -4, 12;
    -5, 9, 10, 2;
    6, 13, 8, 11;
    15, 5, 4, 1];

%
 ************************************************************************
% Part a
disp("Part a" + newline);

% Sort each column and put the results in array B
B = sort(A, 1);
disp("B =");
disp(B);

%
 ************************************************************************
% Part b
disp("Part b" + newline);

% Sort each row and put the results in array C
C = sort(A, 2);
disp("C =");
disp(C)

%
 ************************************************************************
% Part c
disp("Part c" + newline);

% Add each column an put the results in array D
D = sum(A, 1);
disp("D =");
disp(D);

%
 ************************************************************************
% Part d
disp("Part d" + newline);

% Add each row an put the results in array E
E = sum(A, 2);
disp("E =");
disp(E);
clear
```

```
*********************
Problem 9

Part a

B =
    -5     5    -4     1
     3     7     4     2
     6     9     8    11
    15    13    10    12

Part b

C =
    -4     3     7    12
    -5     2     9    10
     6     8    11    13
     1     4     5    15

Part c

D =
    19    34    18    26

Part d

E =
    18
    16
    38
    25
```

Published with MATLAB® R2017b