Roll No.: A016	Name: Varun Khadayate
Class: B. Tech CsBs	Batch: 1
Date of Experiment: 20-09-2022	Subject: ITWS

Lab 8

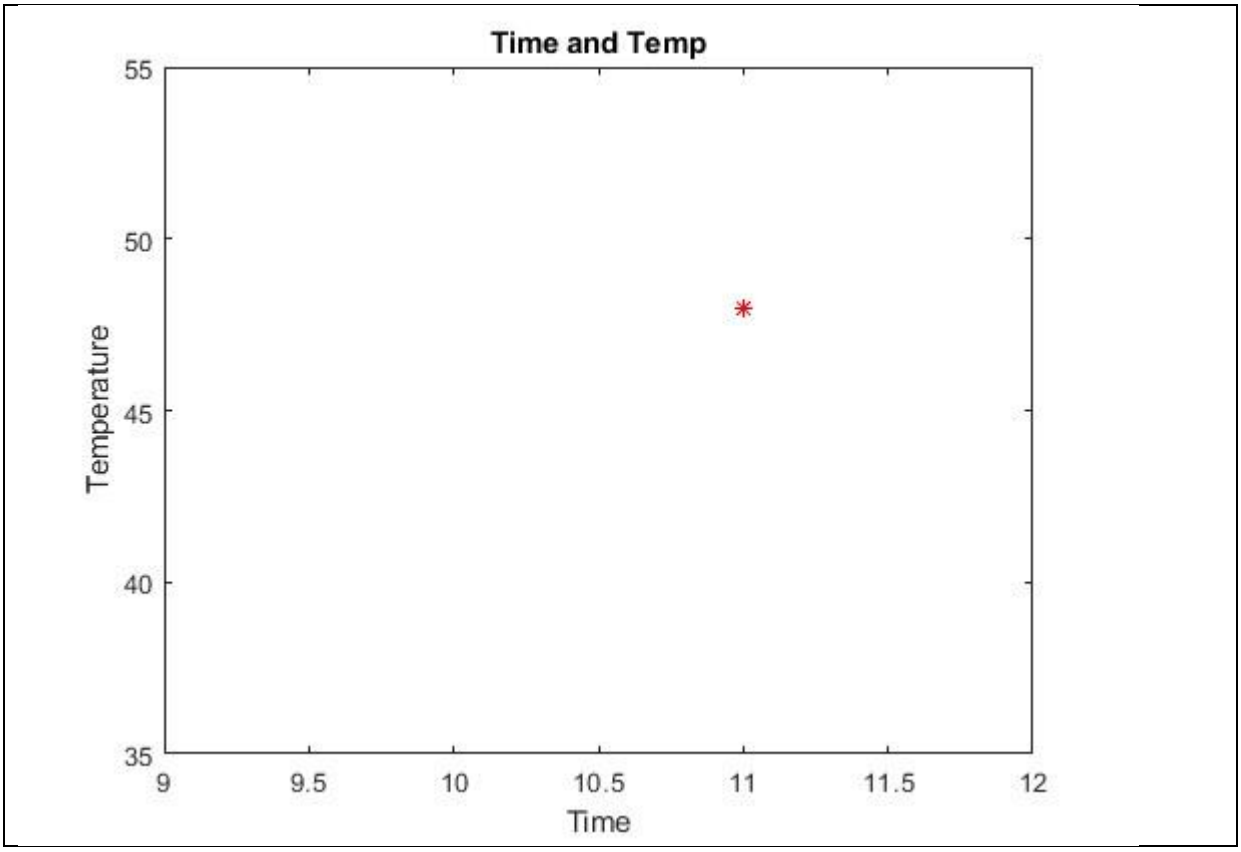
CircleIO.m

```
% This script calculates the area of a circle
% It prompts the user for the radius

%Prompt the user for the radius and calculate the area based on that radius
fprintf('Note: the units will be inches. \n')
radius = input('Please enter the radius: ');
area = pi * (radius ^2) ;

%Print all variables in a sentence format
fprintf('For a circle with a radius of %.2f inches, \n' ,radius)
fprintf('the area is %.2f inches squared \n' , area)
```

```
>> CircleIO
Note: the units will be inches.
Please enter the radius: 5
For a circle with a radius of 5.00 inches,
the area is 78.54 inches squared
```

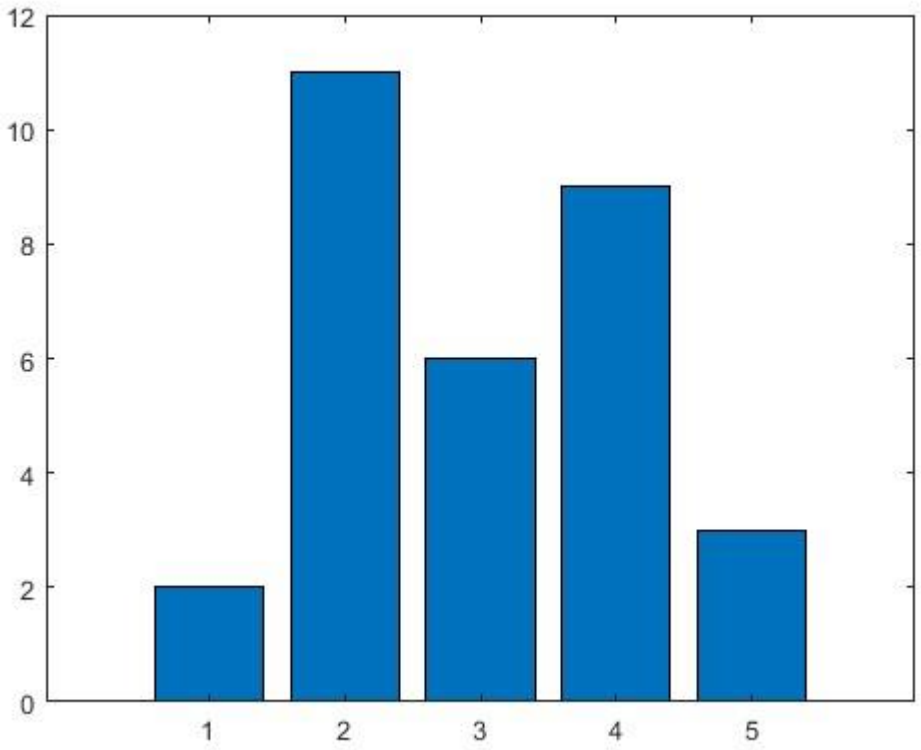
plotonepoint.m

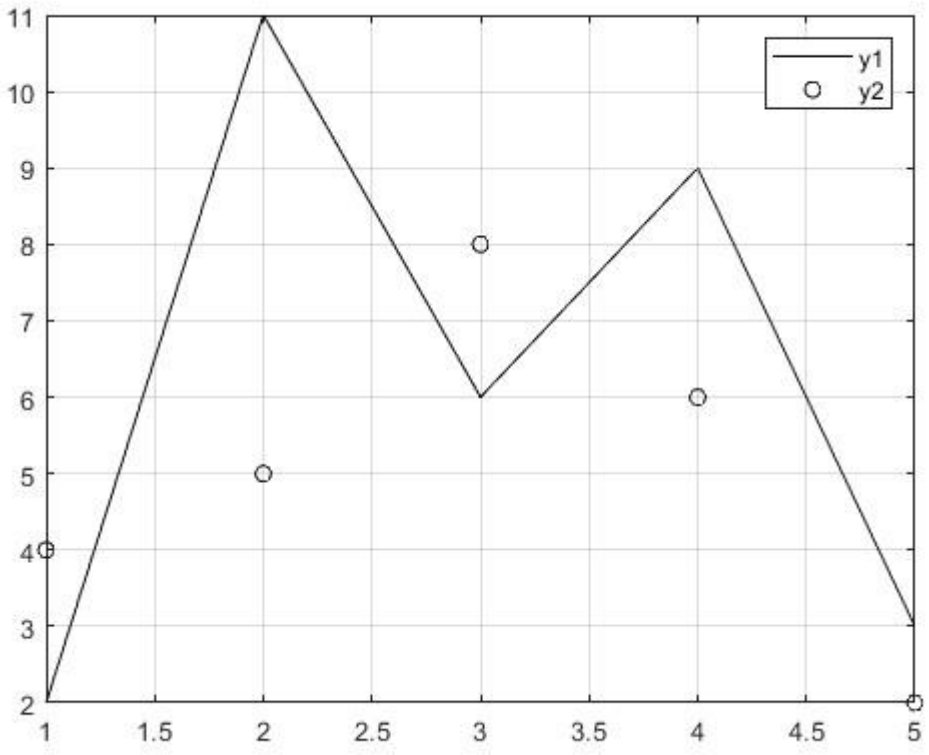
```
%  This is a really simple plot of just one point !
%  Create coordinate variables and plot a red ' *
x = 11;
y = 48;

plot(x, y, 'r*')
%  Change the axes and label them
axis([9 12 35 55])
xlabel('Time')
ylabel('Temperature')
%  Put a title on the plot
title('Time and Temp')
```



plot2figs.m

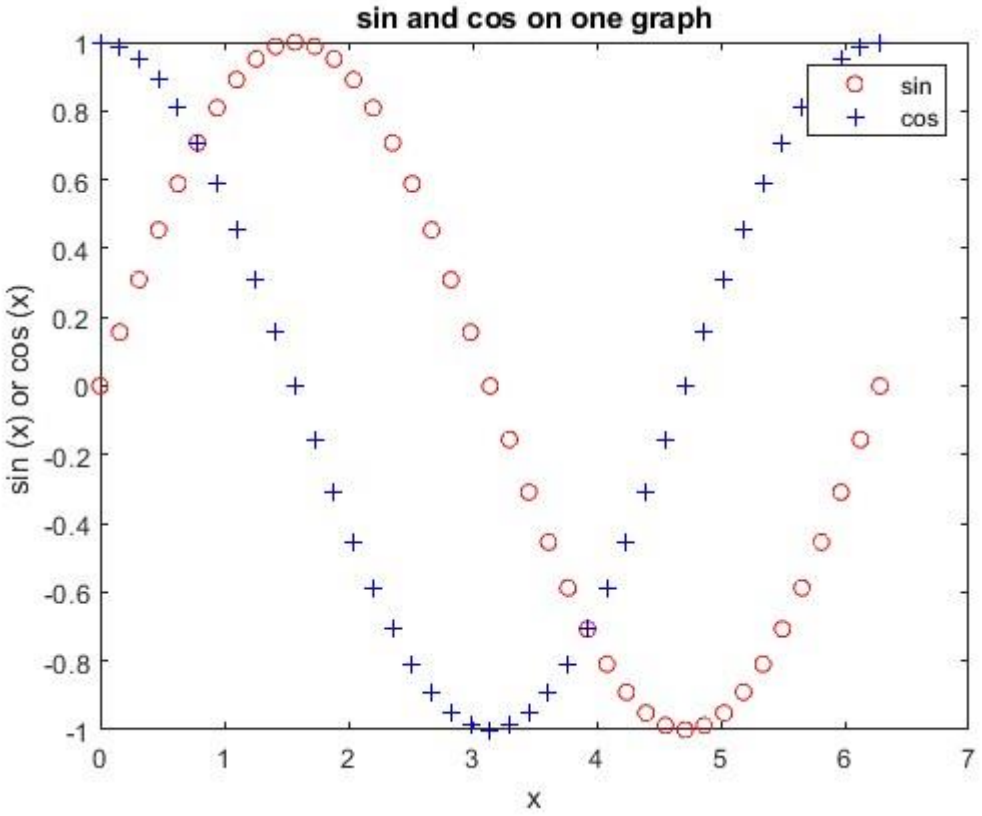
```
% This creates 2 different plots, in 2 different
% Figure Windows, to demonstrate some plot features
clf

x = 1:5; % Not necessary
y1 = [2 11 6 9 3] ;
y2 = [4 5 8 6 2] ;
%  Put a bar chart in Figure 1

figure (1)
bar (x, y1)
%  Put plots using different y values on one plot with a legend
figure (2)

plot (x, y1, 'k')
hold on
plot (x, y2, 'ko' )
grid on
legend ( 'y1' , 'y2')
```


12
10
8
6
4
2
0
1
2
3
4
5


11
10
9
8
7
6
5
4
3
2
1
1.5
2
2.5
3
3.5
4
4.5
5
y1
y2

sinncos.m

```
% takes the character and the number and prints them
a=input('Enter a character--->','s');
n=input('Enter a number--->');
% display the character in a field width of 3.
fprintf('%3c',a);
fprintf('\n')
% display the left-justified number in a field width of 8 with 3 decimal places.
fprintf('%-8.3f\n',n);
```



Practice 3.1

Write a script to calculate the circumference of a circle ($C = 2\pi r$). Comment the script.

```
% This script calculates the circumference of a circle
% It prompts the user for the radius

%Prompt the user for the radius and calculate the area based on that radius
fprintf('Note: the units will be inches. \n')
radius = input('Please enter the radius: ');
area = 2 * pi * radius ;

%Print all variables in a sentence format
fprintf('For a circle with a radius of %.2f inches, \n' ,radius)
fprintf('the Circle is %.2f inches \n' , area)
```

```
>> practice3_1
Note: the units will be inches.
Please enter the radius: 5
For a circle with a radius of 5.00 inches,
the Circle is 31.42 inches
```

Practice 3.2

Create a script that would prompt the user for a length, and then 'f' for feet or 'm' for meters, and store both inputs in variables. For example, when executed it would look like this (assuming the user enters 12.3 and then m):

```
Enter the length: 12.3
Is that f(eet) or m(eters)?: m
```

```
% This script records the length and the unit of that length
% It prompts the user for the radius

%  Promts the user for values and stores them
length=input ( 'Enter the value for length: ') ;
units=input ( 'Enter the units for length (f or m) : ', 's');
```

```
>> practice3_2
Enter the value for length: 12
Enter the units for length (f or m) : m
```

Practice 3.3

Write a script to prompt the user separately for a character and a number, and print the character in a field width of 3 and the number left-justified in a field width of 8 with 3 decimal places. Test this by entering numbers with varying widths.

```
% takes the character and the number and prints them
a=input('Enter a character--->','s');
n=input('Enter a number--->');
% display the character in a field width of 3.
fprintf('%3c',a);
fprintf('\n')
% display the left-justified number in a field width of 8 with 3 decimal places.
fprintf('%-8.3f\n',n);
```

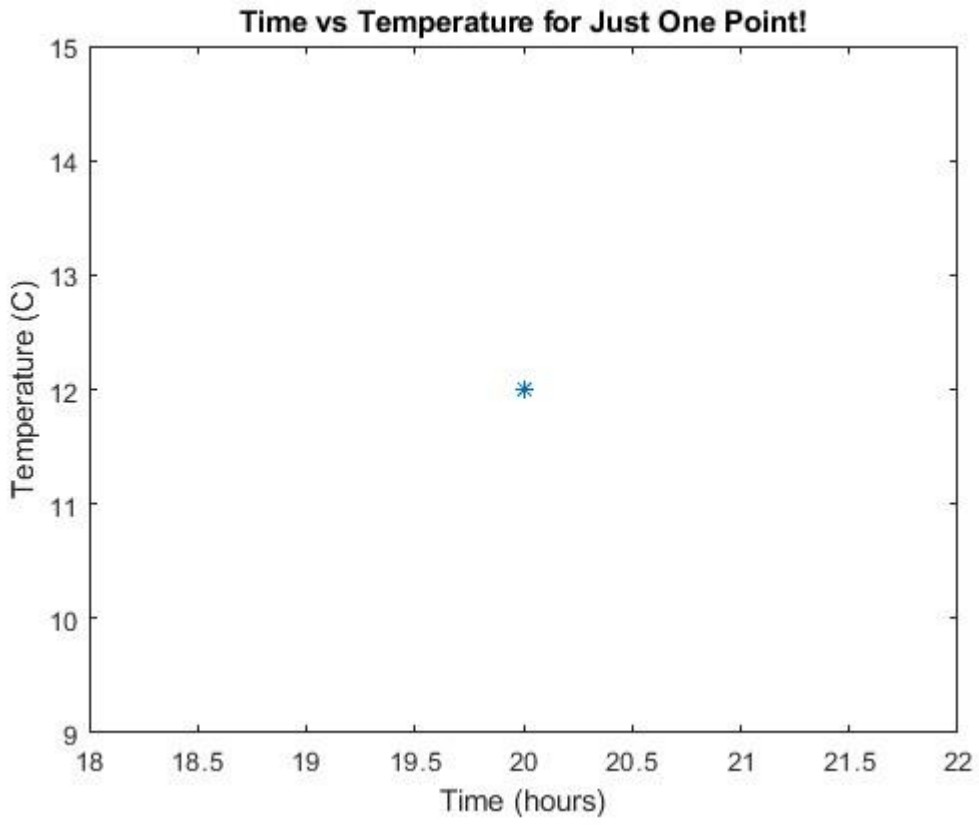
```
>> practice3_3
Enter a character--->varun
Enter a number--->123455678.123344556789
  v  a  r  u  n
123455678.123
```

Practice 3.4

Modify the script *plotonepoint* to prompt the user for the time and temperature, and set the axes based on these values.

```
% This script plots one point (time, temp) from data
% acquired from the user.
% Ask the user for the data to plot and store in the
% variables time and temp
time = input('Enter the time in hours: ');

temp = input('Enter the temperature in degrees C: ');
plot(time,temp,"*")

% Adjust the x and y axis limits

axis([time-2 time+2 temp-3 temp+3]);

% Label the axis
xlabel("Time (hours)")

ylabel("Temperature (C)")
% Add a title
title("Time vs Temperature for Just One Point!")
```

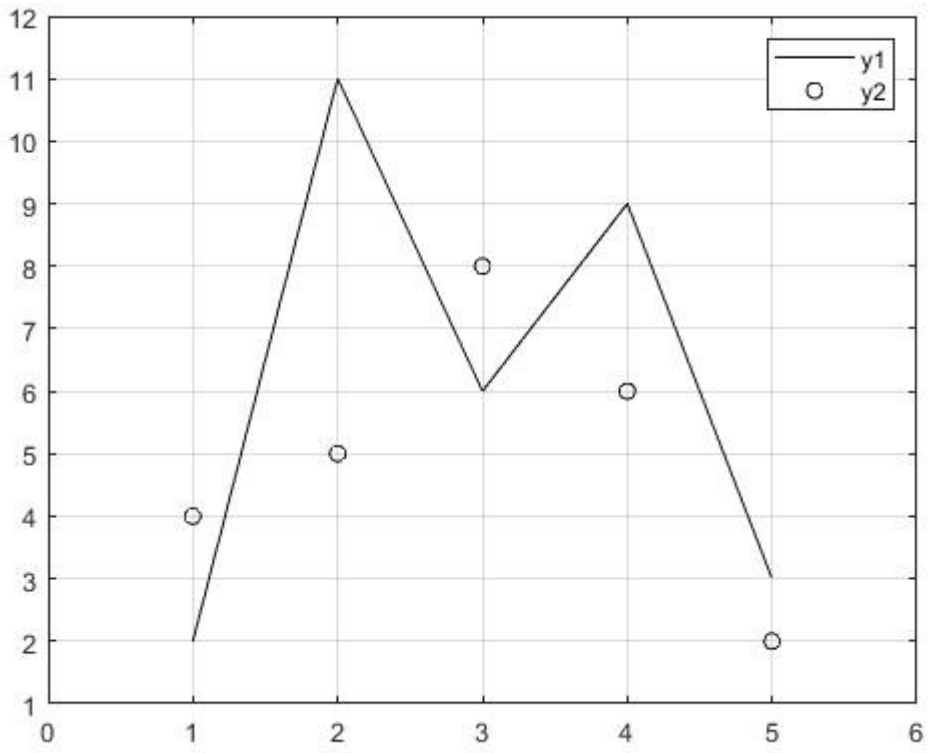
```
>> practice3_4
Enter the time in hours: 12
Enter the temperature in degrees C: 50
```

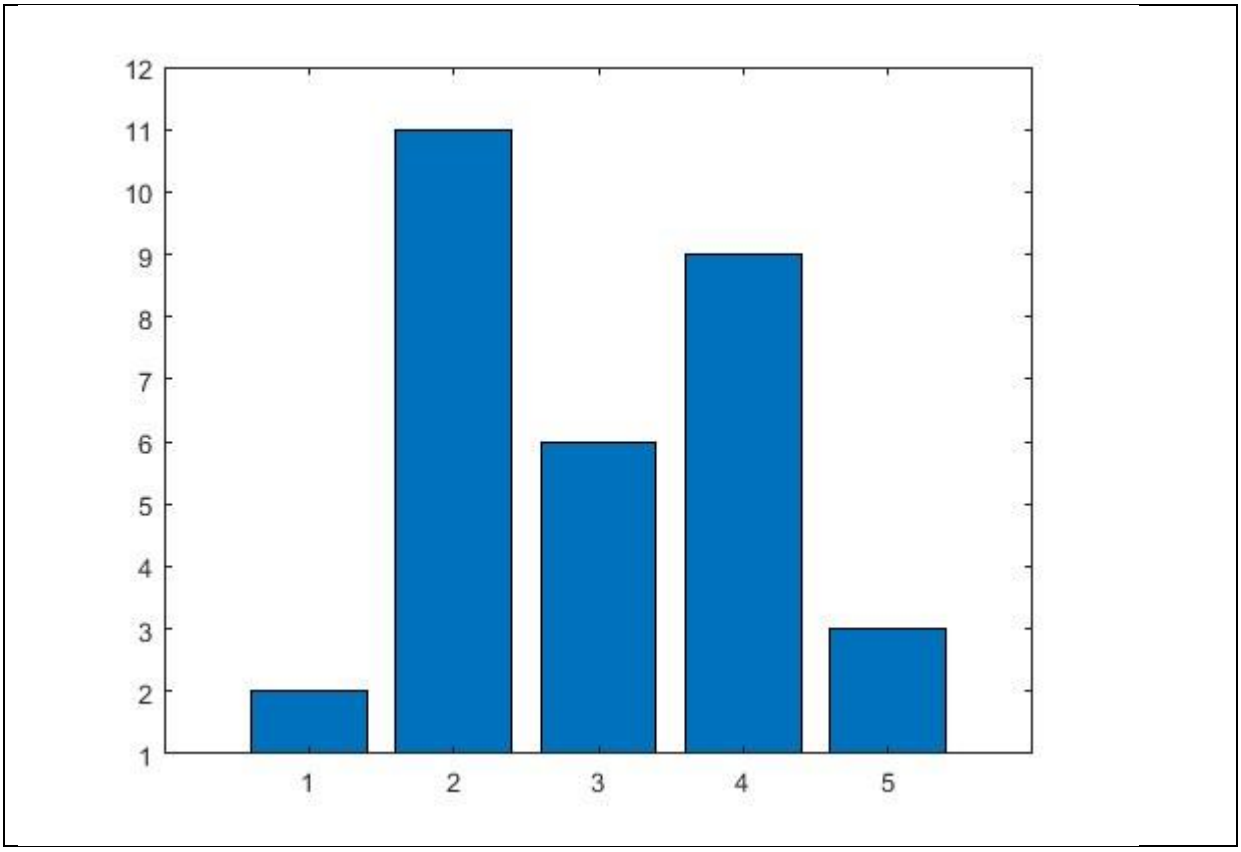


Practice3.5

Modify the *plot2figs* script using the **axis** function so that all points are easily seen.

```
% creates two different plots
x=1:5;
y1=[2 11 6 9 3];

y2=[4 5 8 6 2];
%Put a bar chart in Figure 1
figure(1)
bar(x,y1)
%Change the axis settings
axis([0 6 1 12]);
%Put plots using different y values on one plot with a legend
figure(2)
plot(x,y1,"k")
hold on
plot(x,y2,'ko')
grid on
legend( 'y1', 'y2')
%Change the axis settings
axis([0 6 1 12]);
```

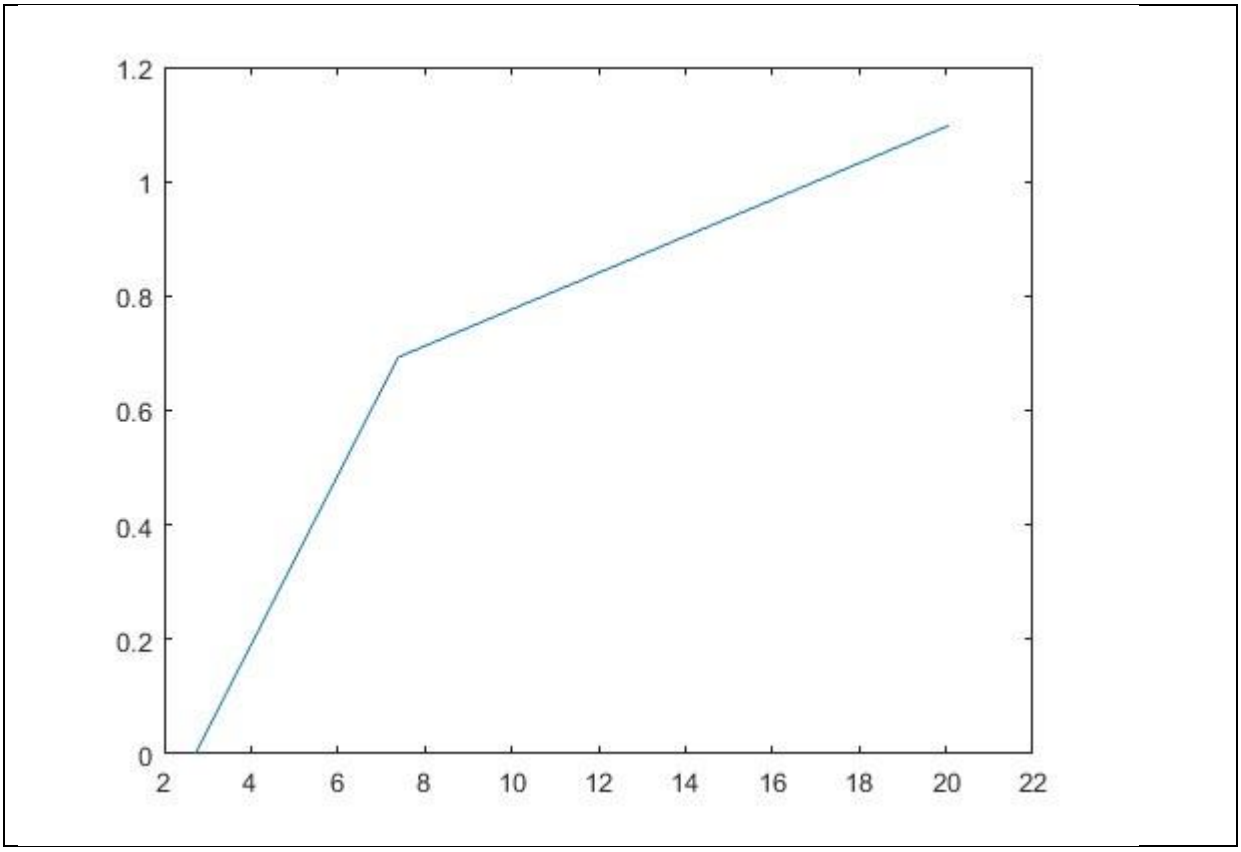




Practice 3.6

Write a script that plots exp(x) and log(x) for values of x ranging from 0 to 3.5.

```
x = 0:3.5;
plot(exp(x),log(x))
```



Practice 3.7

Prompt the user for the number of rows and columns of a matrix, create a matrix with that many rows and columns of random integers, and write it to a file.

```
% This script asks for user input for the number of rows
% and columns for a matrix. Then a matrix of that size is
% filled with random numbers and save in ASCII format to a

% file.
% Ask user for input
row = input('Enter the number of matrix rows: ');
col = input('Enter the number of matrix columns: ');

% Create the matrix of random numbers of specified size.
mat = randi(25, row,col)
% Save the matrix as an ASCII file.
save myMatrix.dat mat -ascii
```

```
>> practice3_7
Enter the number of matrix rows: 5
Enter the number of matrix columns: 5

mat =

    11     1    19     1    18
    23    22    10     7     8
    20    24    17     2    24
    24    17     5     3     1
    17    19    18    21    11
```

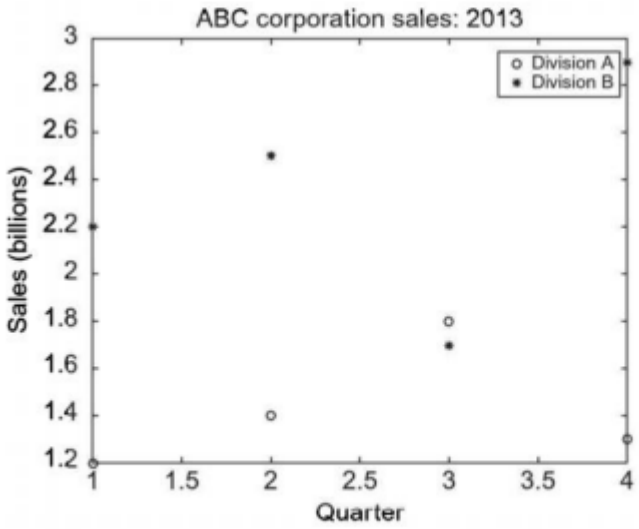
Practice 3.8

The sales (in billions) for two separate divisions of the ABC Corporation for each of the four quarters of 2013 are stored in a file called "salesfigs.dat":

```
1.2 1.4 1.8 1.3
2.2 2.5 1.7 2.9
```

- First, create this file (just type the numbers in the Editor, and Save As "salesfigs.dat").
- Then, write a script that will

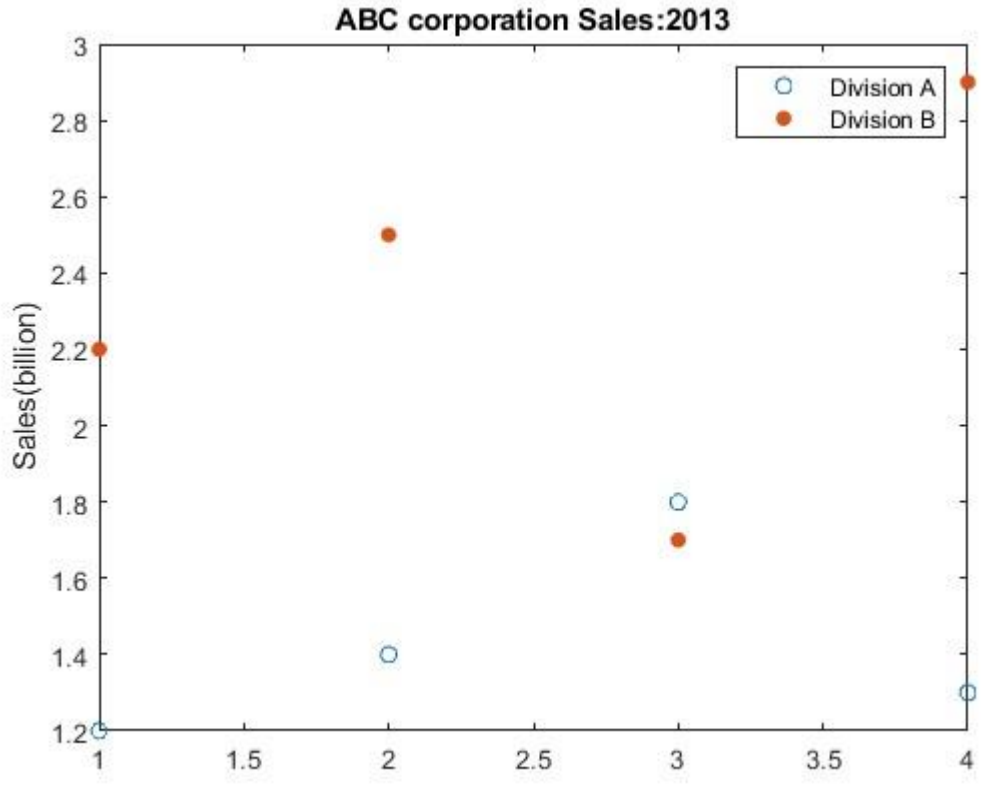
  load the data from the file into a matrix
  separate this matrix into 2 vectors.
  create the plot seen in Fig. 3.7 (which uses black circles and stars as the plot symbols).




```
row = [1.2 1.4 1.8 1.3];
col = [2.2 2.5 1.7 2.9];
mat = [row;col]
% Save the matrix as an ASCII file.
save salesfigs.dat mat -ascii
% Load .dat file
load salesfigs.dat
x=salesfigs(1,:)
y=salesfigs(2,:)
```

```
plot(1:numel(x),x,'o')
hold on
plot(1:numel(y),y,'.','MarkerSize',20)
ylabel('Sales(billion)')
title('ABC corporation Sales:2013')
legend('Division A','Division B')
hold off
```

```
>> practice3_8

mat =

    1.2000    1.4000    1.8000    1.3000
    2.2000    2.5000    1.7000    2.9000


x =

    1.2000    1.4000    1.8000    1.3000


y =

    2.2000    2.5000    1.7000    2.9000
```



Practice 3.9

Write a script that will prompt the user for the radius and height, call the function *conevol* to calculate the cone volume, and print the result in a nice sentence format. So, the program will consist of a script and the *conevol* function that it calls.

```
function [ V ] = conevol( r,h )
% CONEVOL finds the volume of a cone
% Format of call: conevol(r,h)

% Returns cone volume
V = pi * r^2 * h/3;
end
```

```
>> conevol(4,5)

ans =

   83.7758
```

Practice 3.10

For a project, we need some material to form a rectangle. Write a function *calcrectarea* that will receive the length and width of a rectangle in inches as input arguments, and will return the area of the rectangle. For example, the function could be called as shown, in which the result is stored in a variable and then the amount of material required is printed, rounded up to the nearest square inch.

```
>> ra = calcrectarea(3.1, 4.4)
ra =
    13.6400

>> fprintf('We need %d sq in.\n', ceil(ra))
We need 14 sq in.
```

```
function [ A ] = calcrectarea( L,W )
% CALRECTAREA Finds the area of a rectangle rounded up to
% the nearest integer.
% Format of call: calRectArea(L,W)
% Returns rectangle area rounded up to the nearest integer
% Compute the area and round the result up to the nearest integer.
A = ceil( L * W );
end
```

```
>> calcrectarea(4,4)

ans =

    16
```