Dhirubhai Ambani Institute of Information and Communication Technology

ADSP

LAB – 1

Name: Manish Manojkumar Prajapati

Enrolment Number: 202411012

Branch: MTech ICT ML (2024 - 26)

Date: 10/01/2025

EXERCISES

1) A simple sine plot: Plot y=sin(x), $0 \leq x \leq 2\pi$, taking 50 linearly spaced points in the given interval. Label the axes and put 'Plot created by your name' in the title.

CODE: -

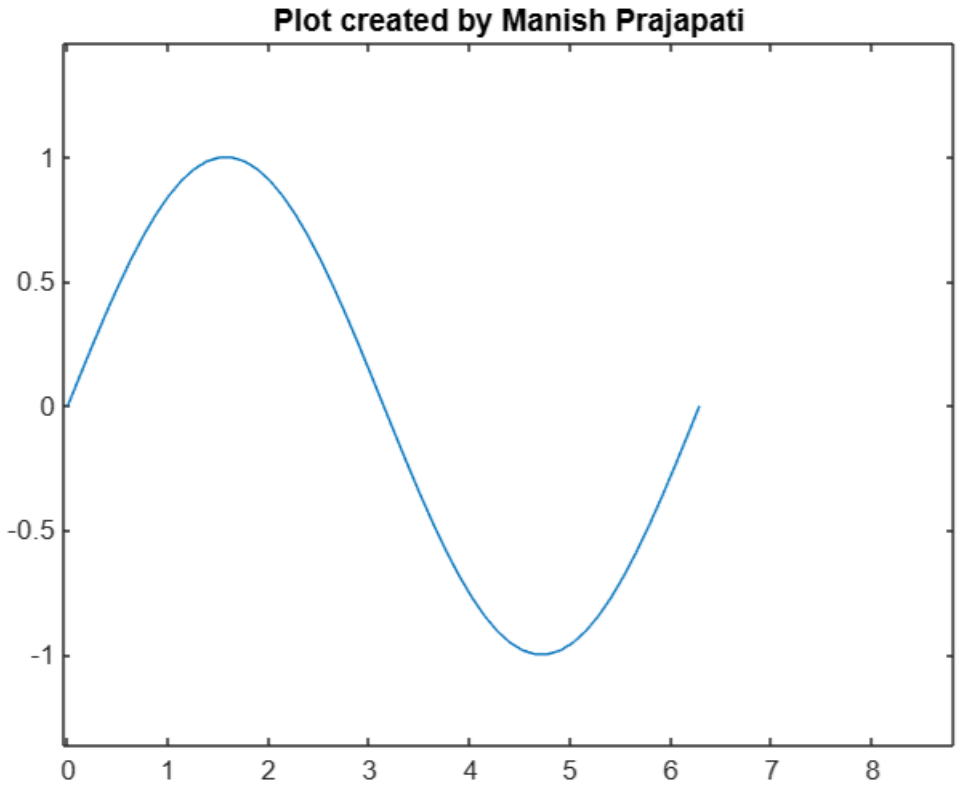
```
>> x = [0 : 2*pi/50 : 2*pi]
```

```
x =

  Columns 1 through 15

         0    0.1257    0.2513    0.3770    0.5027    0.6283    0.7540    0.8796    1.0053    1.1310    1.2566    1.3823    1.5080    1.6336    1.7593

  Columns 16 through 30

    1.8850    2.0106    2.1363    2.2619    2.3876    2.5133    2.6389    2.7646    2.8903    3.0159    3.1416    3.2673    3.3929    3.5186    3.6442

  Columns 31 through 45

    3.7699    3.8956    4.0212    4.1469    4.2726    4.3982    4.5239    4.6496    4.7752    4.9009    5.0265    5.1522    5.2779    5.4035    5.5292

  Columns 46 through 51

    5.6549    5.7805    5.9062    6.0319    6.1575    6.2832
```

```
>> y = sin(x)
```

```
>> y = sin(x)

y =

  Columns 1 through 15

         0    0.1253    0.2487    0.3681    0.4818    0.5878    0.6845    0.7705    0.8443    0.9048    0.9511    0.9823    0.9980    0.9980    0.9823

  Columns 16 through 30

    0.9511    0.9048    0.8443    0.7705    0.6845    0.5878    0.4818    0.3681    0.2487    0.1253    0.0000   -0.1253   -0.2487   -0.3681   -0.4818

  Columns 31 through 45

   -0.5878   -0.6845   -0.7705   -0.8443   -0.9048   -0.9511   -0.9823   -0.9980   -0.9980   -0.9823   -0.9511   -0.9048   -0.8443   -0.7705   -0.6845

  Columns 46 through 51

   -0.5878   -0.4818   -0.3681   -0.2487   -0.1253   -0.0000
```

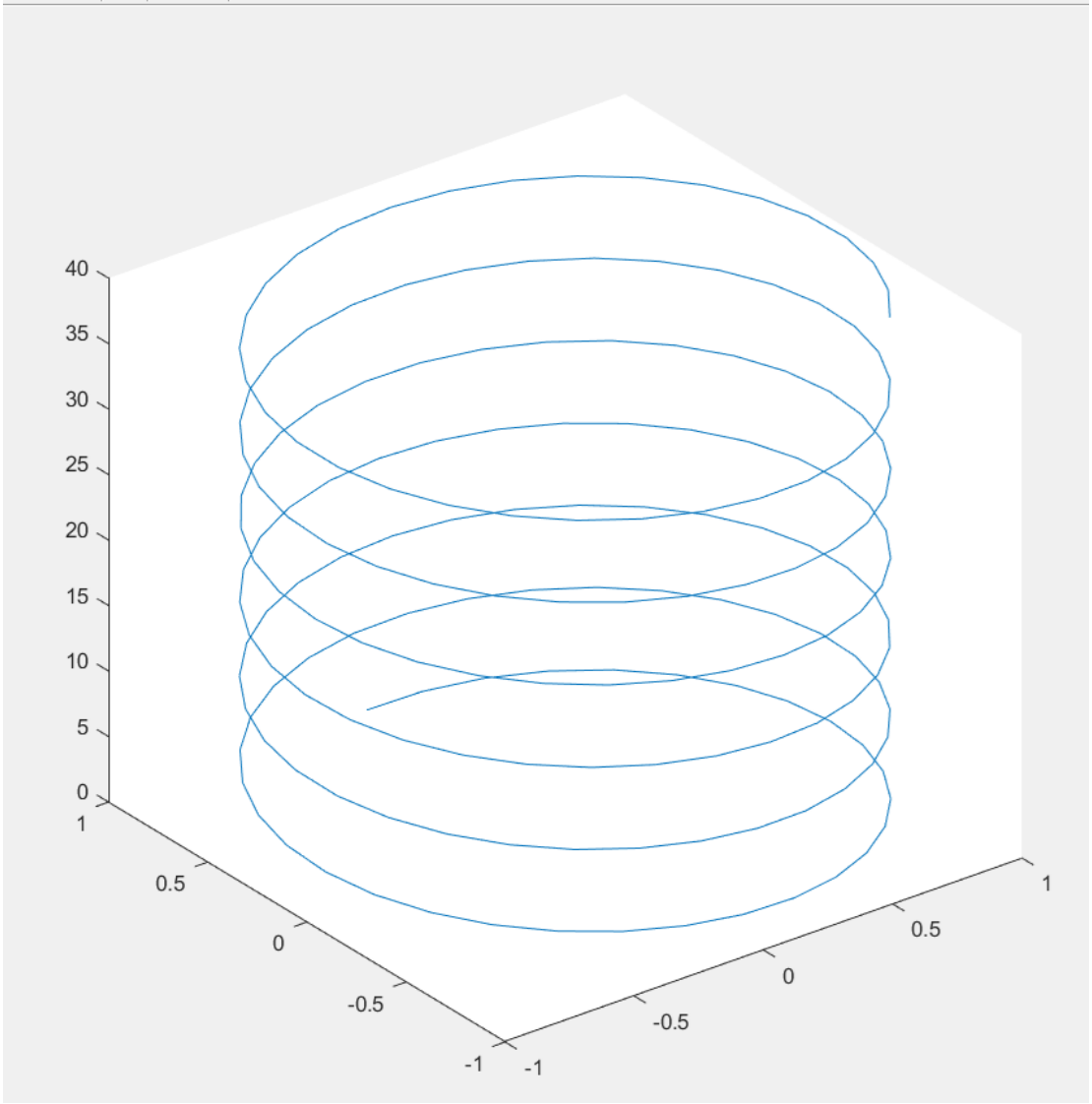
```
>> plot(x, y);

    title('Plot created by Manish Prajapati');
```



2) Use command plot3(x, y, z) to plot the circular helix x(t) = sin(t), y(t) = cos(t), z(t) = t, 0 ≤ t ≤ 40.

CODE: -

```
% Initializing values in t

% Taking 200 values between 0 and 40

t = linspace(0, 40, 200);

% Initializing x, y and z as described in question
x = sin(t);
y = cos(t);
z = t;

% Plotting helix
plot3(x, y, z);
```



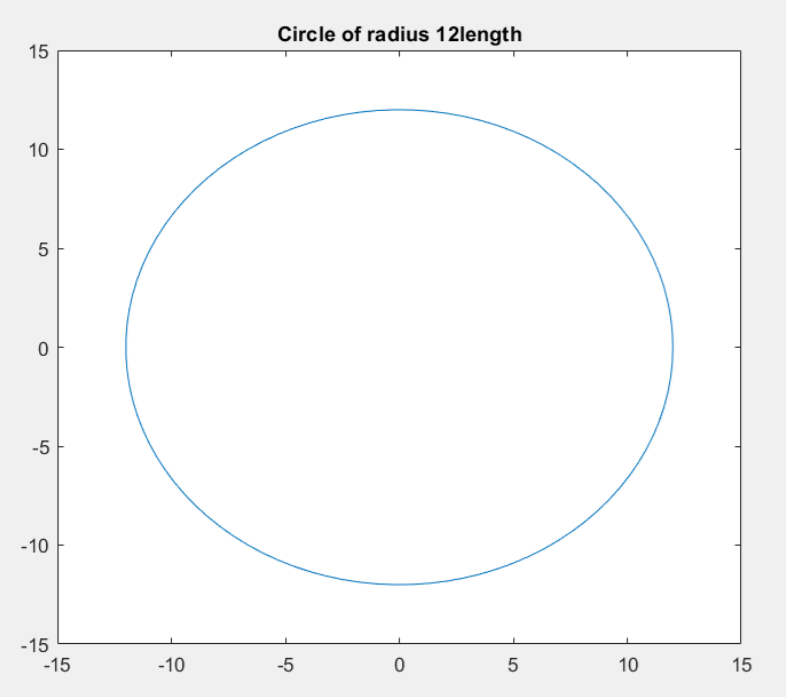
EXERCISE

Question: - Use the command *input* to ask the user to input (r) on the screen in the above program.

CODE: -

```
% This file plots a circle by taking input of radius from the user

clc;
clear all;
close all;
radius = input("Enter radius: ");


% Initializing degrees, coordinates and all
theta = linspace(0, 2*pi, 100);
x = radius*cos(theta);
y = radius*sin(theta);

title_value = 'Circle of radius ' + string(radius) + 'length';

% Plotting circle
plot(x, y);
title(title_value);
```

```
Command Window
New to MATLAB? See resources for Getting Started.
   Enter radius: 12
fx >>
```

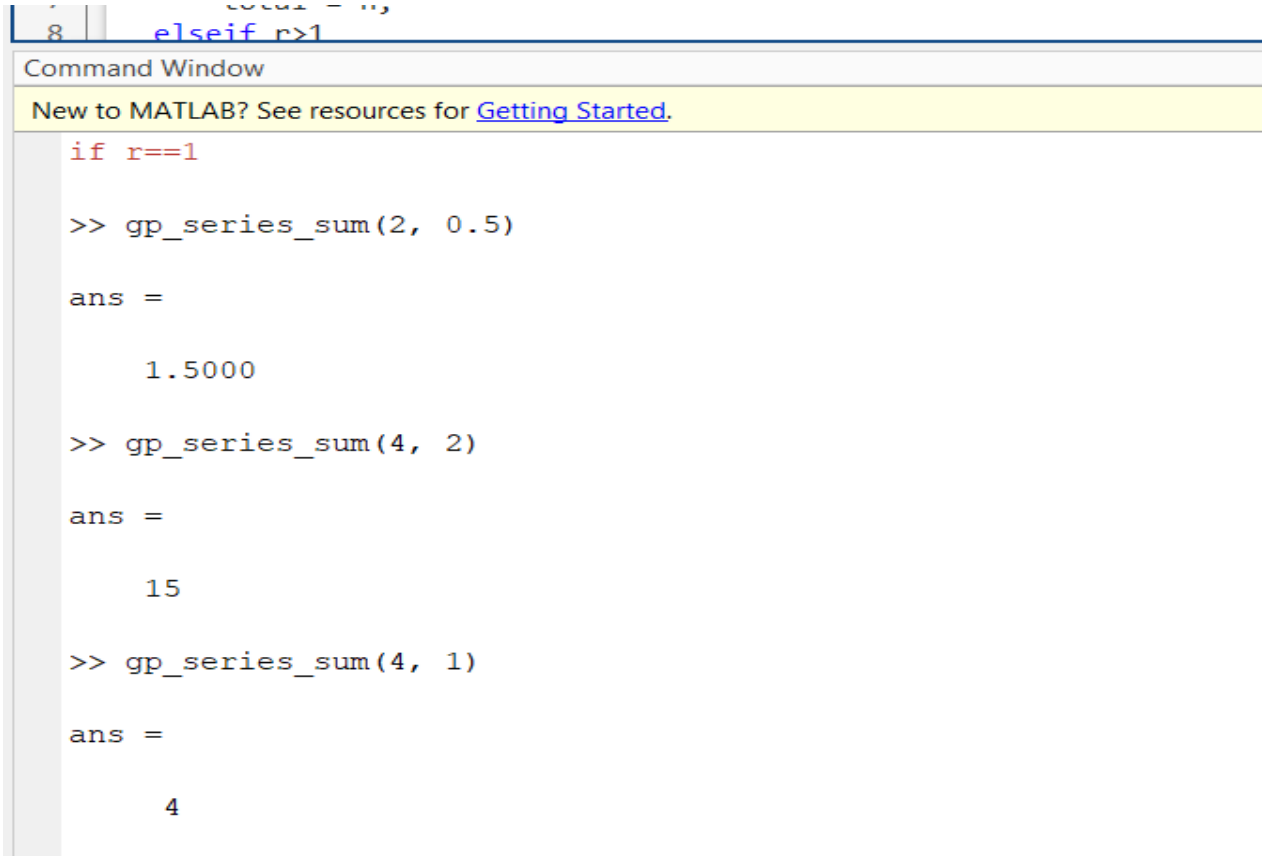


EXERCISE – 1

- Write a function to compute the sum of a geometric series $1 + r + r^2 .. + r^n$ for given r and n.

CODE: -

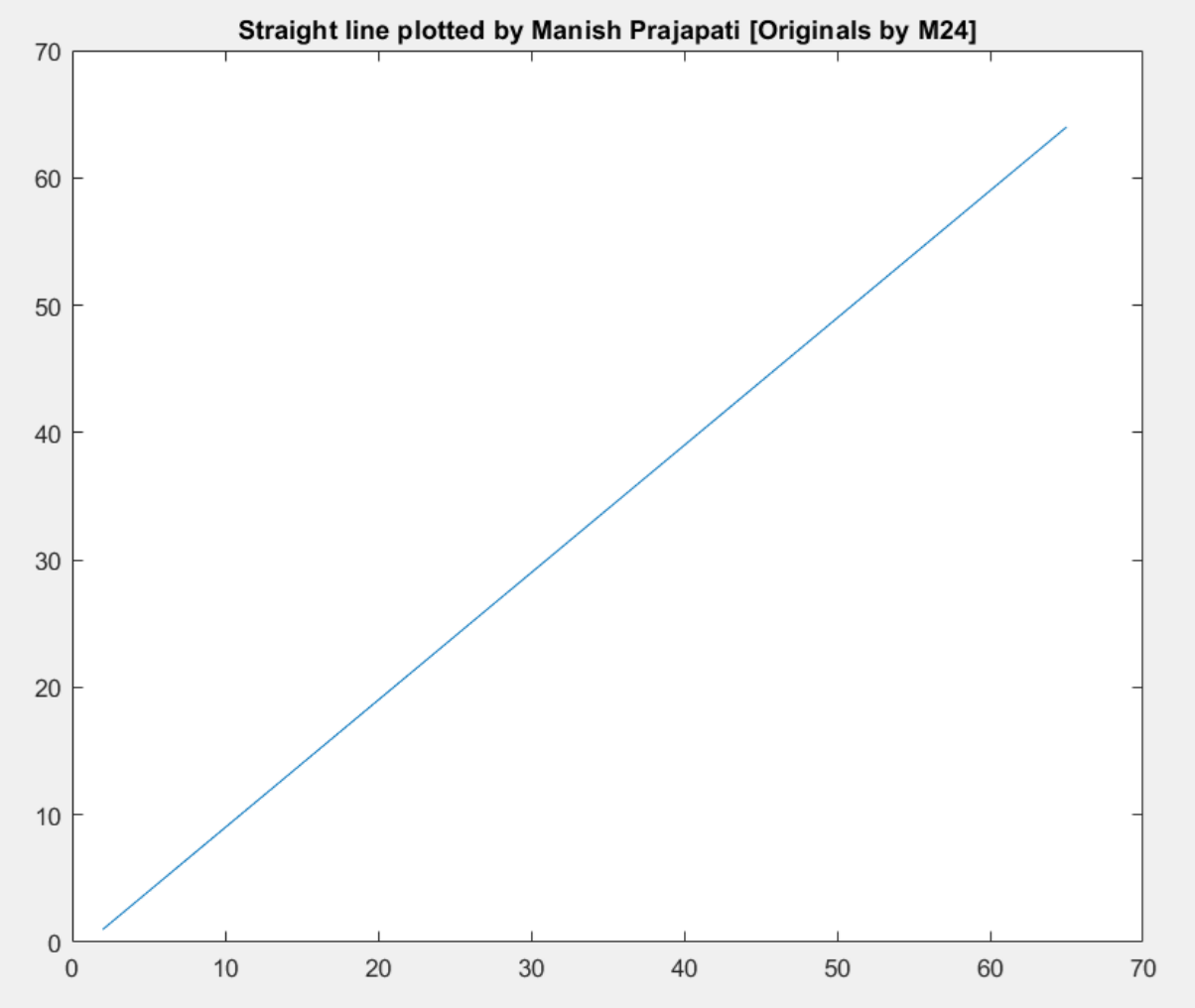
```
function total = gp_series_sum(n, r)
% gp_series_sum will calculate the sum total of series:
%  1 + r + r^2 + r^3 + . . . .
% where r = ratio and n = number of terms

if r==1
    total = n;
elseif r>1
    total = 1 * (r^n - 1)/(r - 1);
elseif r<1
    total = 1 * (1 - r^n) / (1 - r);
end

end
```



- Plot a line y = mx + c by taking 6 different values of x. Take m=1 and c=-1.

CODE: -

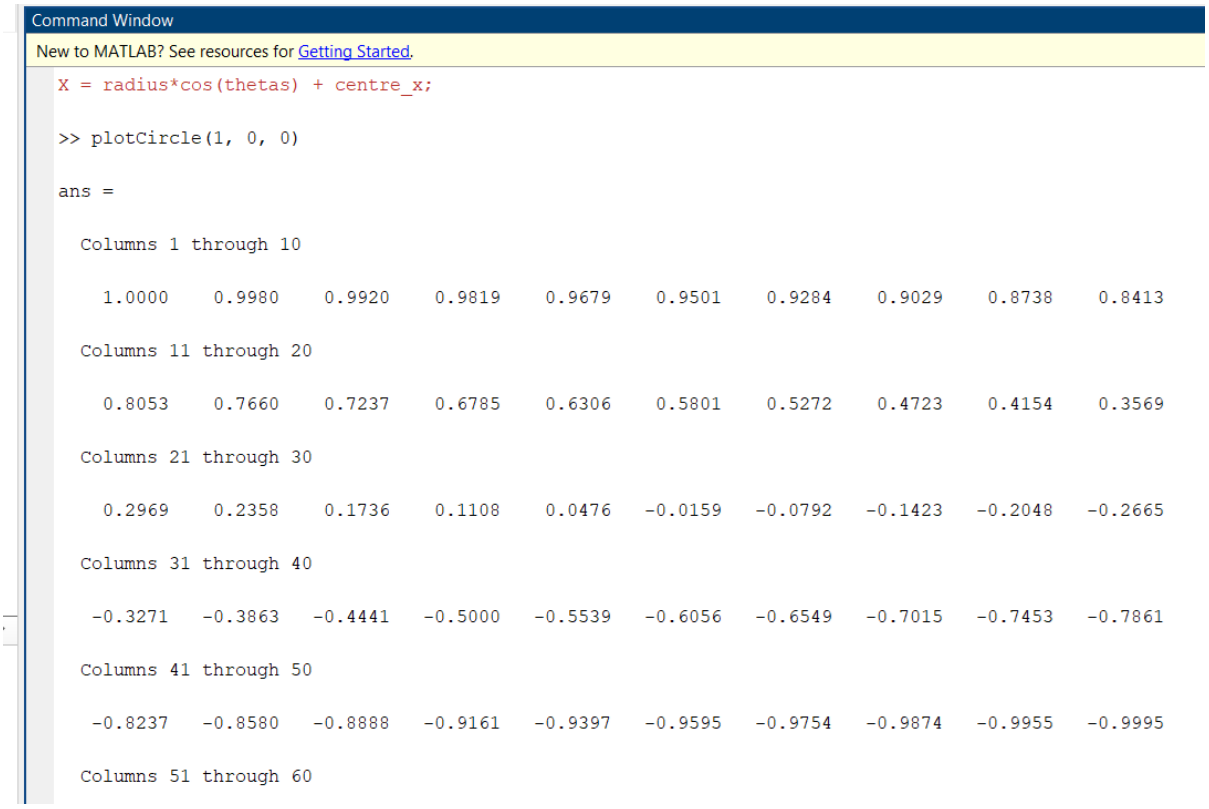
```
% Plotting a straight line y = mx + c
% where x will have different values and
% m=1 and c=-1 is taken

x = [2 12 19 59 65];
m = 1;
c=-1;

y = m*x +c;

plot(x, y);
title("Straight line plotted by Manish Prajapati [Originals by M24]");
```

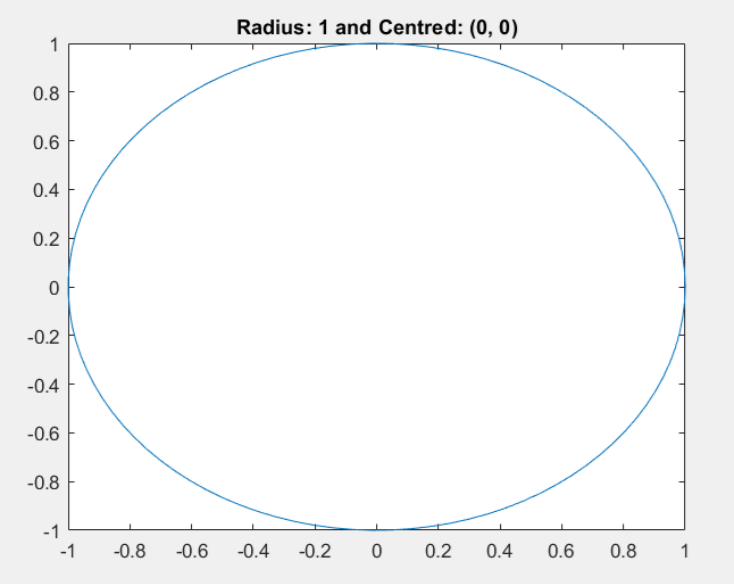


- Plot a circle of unit radius. Hint: - Use parametric (polar) coordinate system. Generalize this to any centre and any radius circle.

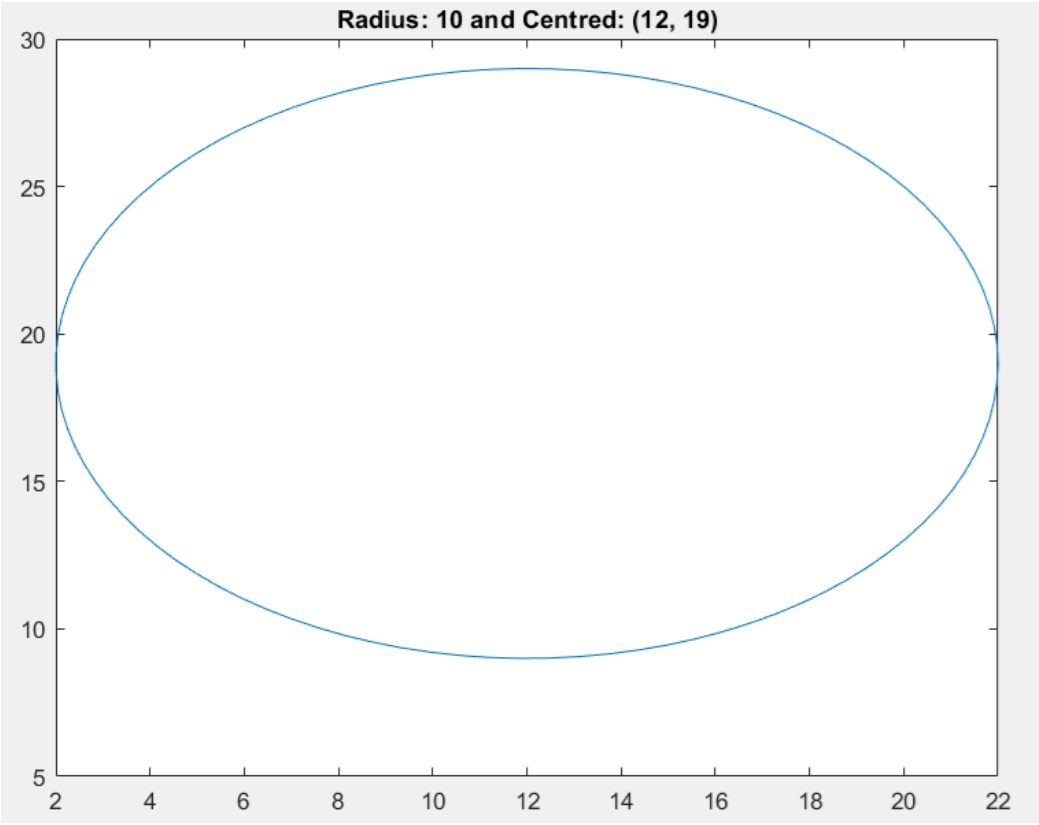
CODE: -

```
function [X, Y] = plotCircle(radius, centre_x, centre_y)

% plotCircle function takes input arguments: radius, centre_x, centre_y
% centre_x and centre_y is to move the circle's centre away from default (0, 0)

thetas = linspace(0, 2*pi, 100);

X = radius*cos(thetas) + centre_x;
Y = radius*sin(thetas) + centre_y;

title_text = "Radius: " + string(radius) + " and Centred: (" + string(centre_x) +
", " + string(centre_y) + ")";

plot(X, Y);
title(title_text);

end
```






```
>> [X, Y] = plotCircle(10, 12, 19)

X =

  Columns 1 through 10

   22.0000   21.9799   21.9195   21.8193   21.6795   21.5007   21.2837   21.0293   20.7385   20.4125

  Columns 11 through 20
```



- Using Taylor series expansion find the value of e^2. How many terms in the expansion do you need to find the exact value with 0.0001 precision?

CODE: -

```
% This program is the approximation of value of e^2
% Here, I have generalized this program to give approximate value of
% e^x, where x>=0

% Taylor series will be used to find this value
% Precision: 0.0001

x = input("Enter value of x for e^x: ");

precision = 0.0001;

i = 0;
e_x = 0;

while true

    last_term = x^i/factorial(i);

    if last_term<precision
        break
    else
        e_x = e_x + last_term;
    end

    i = i+1;

end

disp("Iterations: " + string(i));
disp("x: " + string(x));
disp("e_x: " + string(e_x));
```

```
>> Lab_1_exercise_1_e2_approximation
Enter value of x for e^x: 2
Iterations: 11
x: 2
e_x: 7.389
>> Lab_1_exercise_1_e2_approximation
Enter value of x for e^x: 1
Iterations: 8
x: 1
e_x: 2.7183
>> Lab_1_exercise_1_e2_approximation
Enter value of x for e^x: 3
Iterations: 14
x: 3
e_x: 20.0855
>> Lab_1_exercise_1_e2_approximation
Enter value of x for e^x: 12
Iterations: 39
x: 12
e_x: 162754.7913
```

- dftmtx.m function is used to get DFT matrix. Read help of this function and find 4 point DFT matrix. Now, verify the DFT matrix is an orthogonal matrix.

CODE: -

```
x = 1:4;

n = length(x);

y = x*dftmtx(n);

disp("n = " + string(n));

disp(y);

y_orth = orth(y);

disp("y_orth: " + string(y_orth));

if y_orth ==1
    disp("DFT for " + string(n) + " data points forms Orthogonal Matrix.");
else
    disp("Orthogonal Matrix is not formed after DFT transformation.");
end
```

```
>> Lab_1_exercises_1_dft
n = 4
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

y_orth: 1
DFT for 4 data points forms Orthogonal Matrix.
>>
```

EXERCISE – 2

- Read a wave file in MATLAB. Observe sampling frequency and bits required to encode one sample, i.e., bit depth and data rate. Can you predict the size of a wave file? Now, increase or decrease the amplitude and write this new wave file as another name (say, output.wav)