MATLAB IMPORTANT PROBLEMS WITH SOLUTIONS

By - Ayushi Singh
ayushi4615@gmail.com

INDEX

14	**WAP in MATLAB to generate a few activation functions that are being used in neural network**		
15	**WAP in MATLAB to perform Fuzzy set Operations.**		
16	**WAP in MATLAB to create** **(a) Create a matrix of zeros with 2 rows and 4 columns.** **(b) Create the row vector of odd numbers through 21, L = 1 3 5 7 9 11 13 15 17 19 21 Use the colon operator.** **(c) Find the sum S of vector L’s elements.** **(d) Form the matrix A = 2 3 2 1 0 1.**		
17	**WAP in MATLAB to create** **(a) Create two different vectors of the same length and add them.** **(b) Now subtract them.** **(c) Perform element-by-element multiplication on them.** **(d) Perform element-by-element division on them.** **(e) Raise one of the vectors to the second power.** **(f) Create a 3 × 3 matrix and display the first row of and the second column on the screen.**		
18	**WAP in MATLAB Using the plot command for multiple plots, plot y = sin(x) and y = cos(x) on the same graph for values of x defined by: x = 0: pi/30:2*pi.**		
19			

	WAP in MATLAB Using the plot command for a single plot and the hold commands, plot y = sin(x) and y = cos(x) on the same graph for values of x defined by: x = 0: pi/30:2*pi.		
20	**(a) write a program in MATLAB that reads an input temperature in degree Fahrenheit, converts it to an absolute temperature in kelvin. Hint: TK = [5/9 (TF-32) +273], TF = 97;** **(b) Write a program to find out the distance between two points (x1, y1) and (x2,y2) specified by the cartesian coordinate plane. Hint: d = sqrt $(x1-x2)^2 + (y1-y2)^2$.**		
21	**Suppose that x=3 and y=4. Use the MATLAB to evaluate the following expression:** **(a) $W = x^2y^2 / (x-y)^2$** **(b) $Z = 2 / y^2(x+y)^2$**		
22	**Write the following expression in MATLAB** **(1) x = 4u / 3v** **(2) $y = v^3 2\pi / v^3-u^3$** **(3) z = √ $5e^{-0.2t}$ cos 2t**		
23	**Write a MATLAB program for perceptron net for an AND function with bipolar inputs and targets.**		

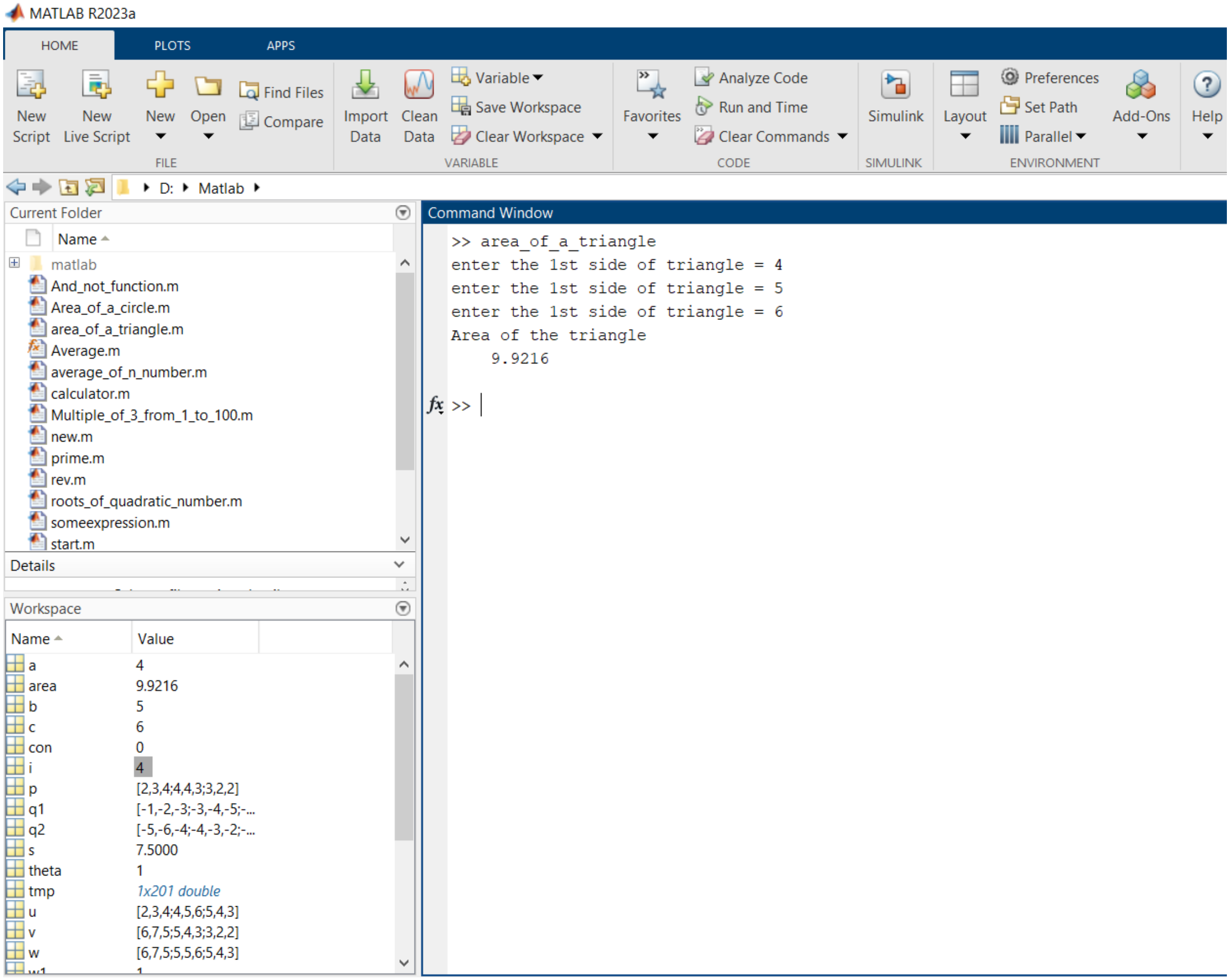
PROGRAM-1

Aim - WAP in MATLAB to find the Area of Triangle.

SOURCE CODE:

```
%Taking input of sides of a triangle

a = input ('enter the 1st side of triangle = ');
b = input ('enter the 1st side of triangle = ');
c = input ('enter the 1st side of triangle = ');

%Calculating the semi perimeter of a triangle
s = (a + b + c) ./ 2 ;

% Calculating the area of a triangle

area = sqrt (s * (s - a) * (s - b) * (s - c));

% Displaying the area of triangle

disp ('Area of the triangle')
disp (area)
```

OUTPUT:



matlab
And_not_function.m
Area_of_a_circle.m
area_of_a_triangle.m
Average.m
average_of_n_number.m
calculator.m
Multiple_of_3_from_1_to_100.m
new.m
prime.m
rev.m
roots_of_quadratic_number.m
someexpression.m
start.m

Name	Value
a	4
area	9.9216
b	5
c	6
con	0
i	4
p	[2,3,4;4,4,3;3,2,2]
q1	[-1,-2,-3;-3,-4,-5;-...
q2	[-5,-6,-4;-4,-3,-2;-...
s	7.5000
theta	1
tmp	1x201 double
u	[2,3,4;4,5,6;5,4,3]
v	[6,7,5;5,4,3;3,2,2]
w	[6,7,5;5,5,6;5,4,3]
w1	1

```
>> area_of_a_triangle
enter the 1st side of triangle = 4
enter the 1st side of triangle = 5
enter the 1st side of triangle = 6
Area of the triangle
    9.9216

fx >> 
```

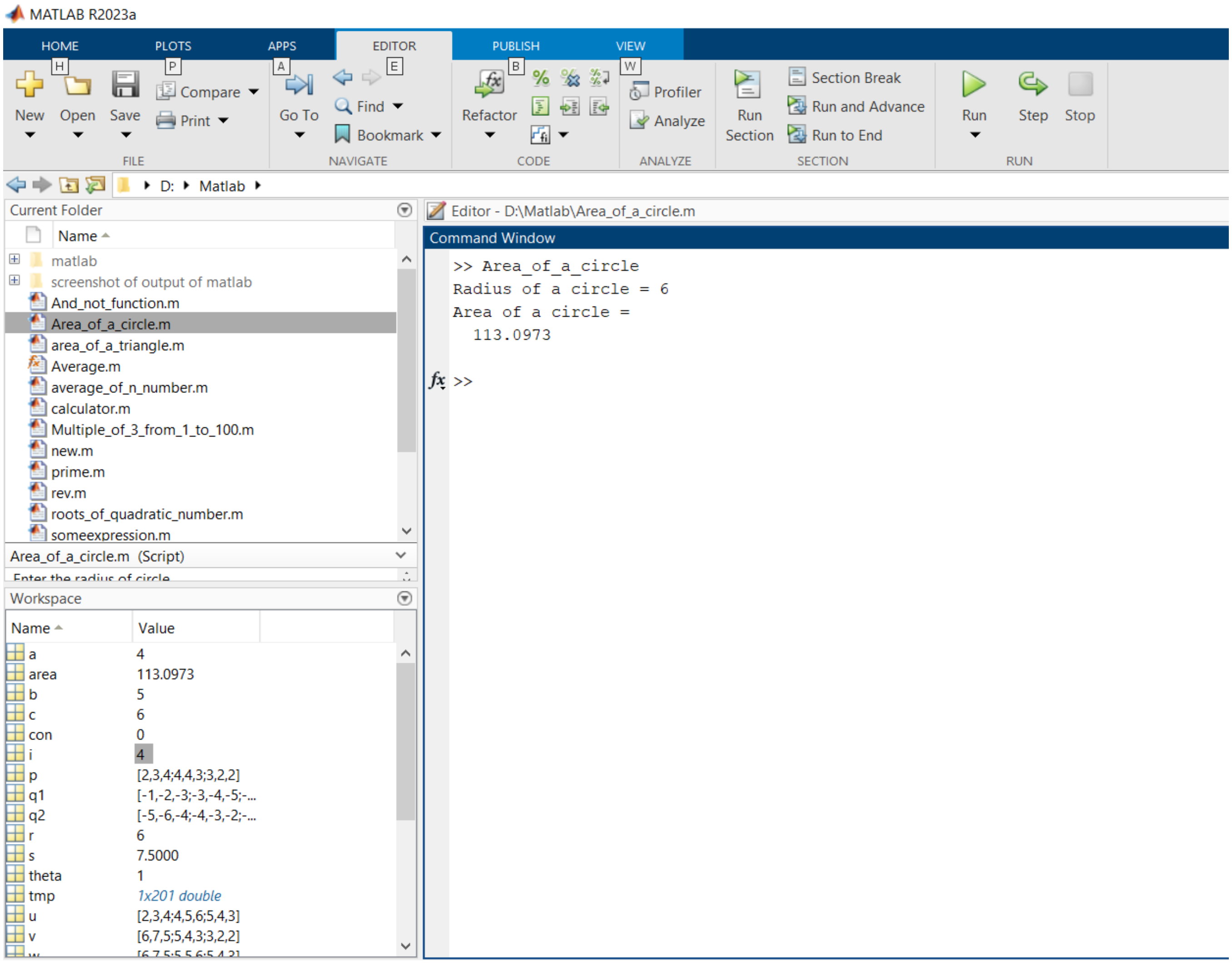
PROGRAM-2

Aim - WAP in MATLAB to find the Area of Circle

SOURCE CODE:

```
%Enter the radius of circle

r = input('Radius of a circle = ');

%Calculate area of a circle

area = pi * r * r ;

%Display the Area of a circle

disp('Area of a circle = ')
disp(area)
```

OUTPUT :



- matlab
- screenshot of output of matlab
- And_not_function.m
- Area_of_a_circle.m
- area_of_a_triangle.m
- Average.m
- average_of_n_number.m
- calculator.m
- Multiple_of_3_from_1_to_100.m
- new.m
- prime.m
- rev.m
- roots_of_quadratic_number.m
- someexpression.m

Area_of_a_circle.m (Script)

Enter the radius of circle

```
>> Area_of_a_circle
Radius of a circle = 6
Area of a circle =
  113.0973

fx >>
```

Name	Value
a	4
area	113.0973
b	5
c	6
con	0
i	4
p	[2,3,4;4,4,3;3,2,2]
q1	[-1,-2,-3;-3,-4,-5;-...
q2	[-5,-6,-4;-4,-3,-2;-...
r	6
s	7.5000
theta	1
tmp	1x201 double
u	[2,3,4;4,5,6;5,4,3]
v	[6,7,5;5,4,3;3,2,2]

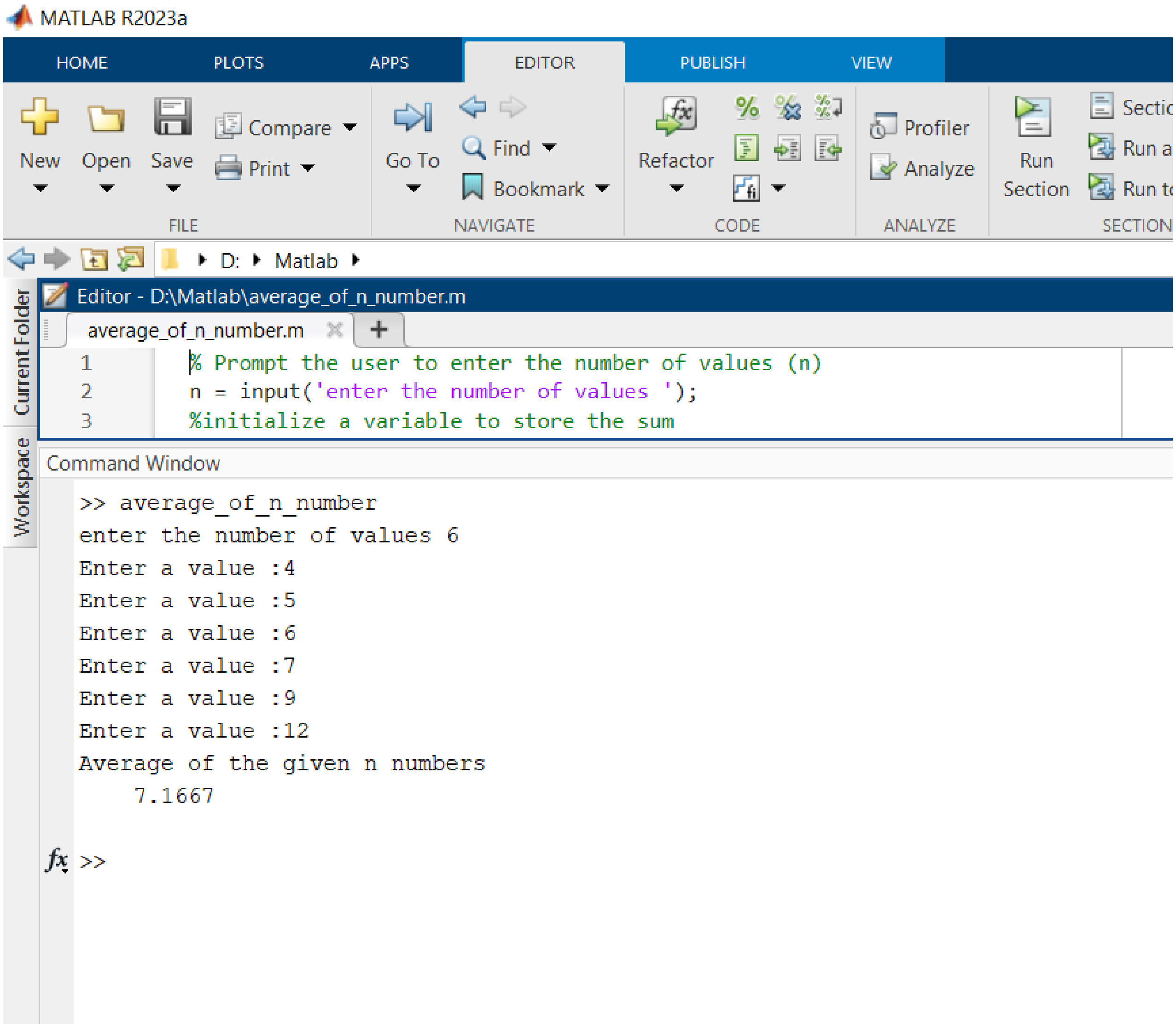
PROGRAM-3

Aim - WAP in MATLAB to find the average of n Numbers.

SOURCE CODE:

```
% Prompt the user to enter the number of values (n)

n = input('enter the number of values ');

% Initialize a variable to store the sum

sum = 0 ;

% Loop to input n number

for i =1:n
value = input ('Enter a value :');
sum = sum + value ;
end

% Calculate the average
aver = sum / n;

% Display the result

disp('Average of the given n numbers ')
disp(aver)
```

OUTPUT:



```
% Prompt the user to enter the number of values (n)
n = input('enter the number of values ');
%initialize a variable to store the sum
```

```
>> average_of_n_number
enter the number of values 6
Enter a value :4
Enter a value :5
Enter a value :6
Enter a value :7
Enter a value :9
Enter a value :12
Average of the given n numbers
    7.1667

>>
```

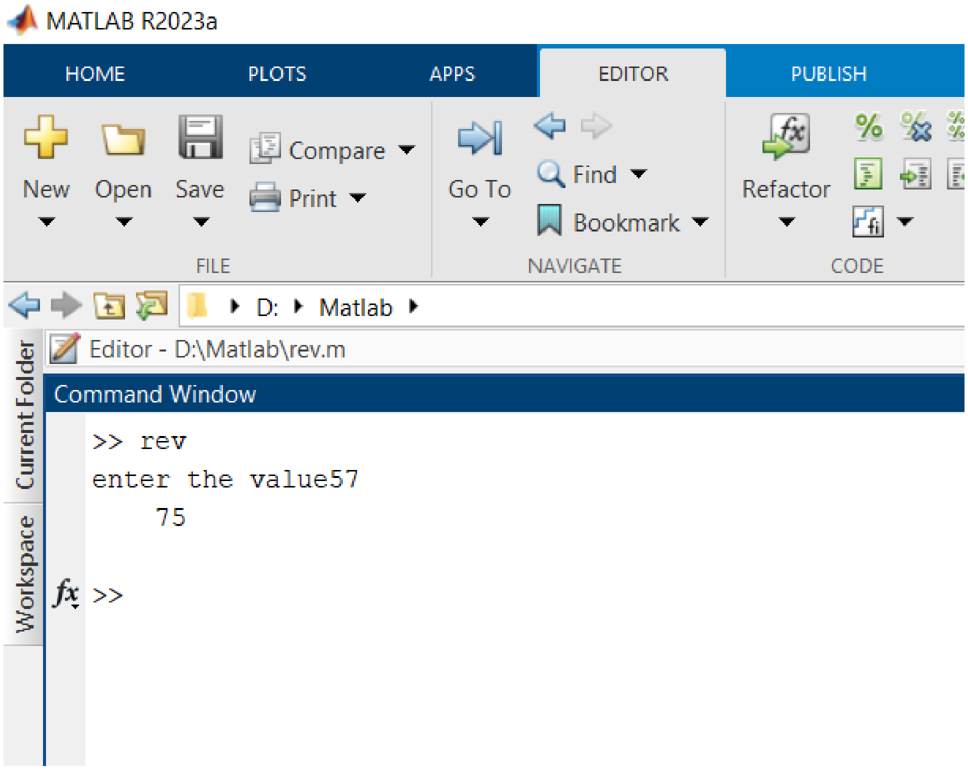
PROGRAM-4

Aim- WAP in MATLAB to Reverse the integer.

SOURCE CODE:

```
% Enter the value that has to be reversed

n = input('enter the value');

% Initialize a variable to store the reverse number

rev_num = 0 ;

% Loop to get reverse of a input number

while n > 0
rem = mod (n , 10);
rev_num = 10 * rev_num+ rem;
n= (n-rem) / 10 ;
end

% Display of the reverse number

disp(rev_num)
```

OUTPUT:



```
>> rev
enter the value57
    75

>>
```

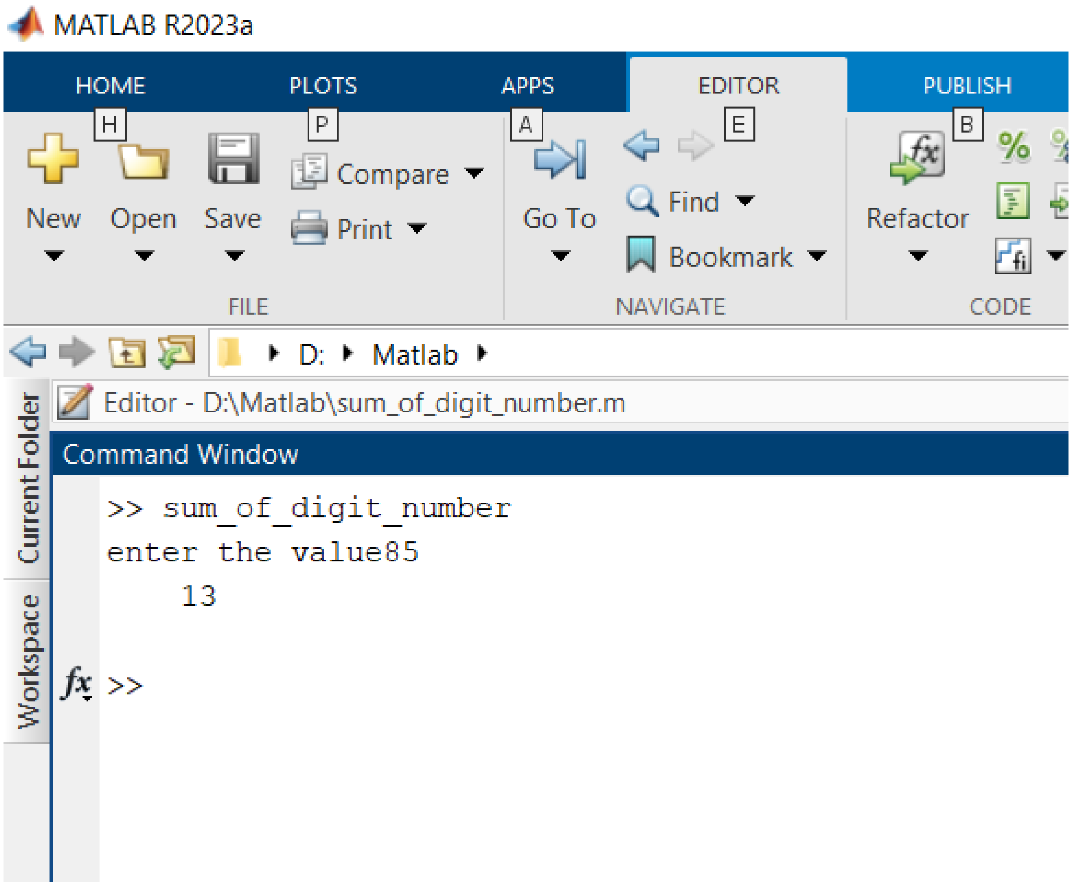
PROGRAM-5

Aim - WAP in MATLAB to find the Sum of digit of a number.

SOURCE CODE:

```
% Enter the value
n = input('enter the value');

% Initialize a variable to store the Sum of digit of a number

sum_num = 0 ;

% Loop to get sum of digit of a input number

while n > 1
rem = mod ( n , 10);
sum_num = rem + sum_num ;
n = (n-rem) /10 ;
end

% Display the result
disp (sum_num)
```

OUTPUT:



```
>> sum_of_digit_number
enter the value85
    13

>>
```

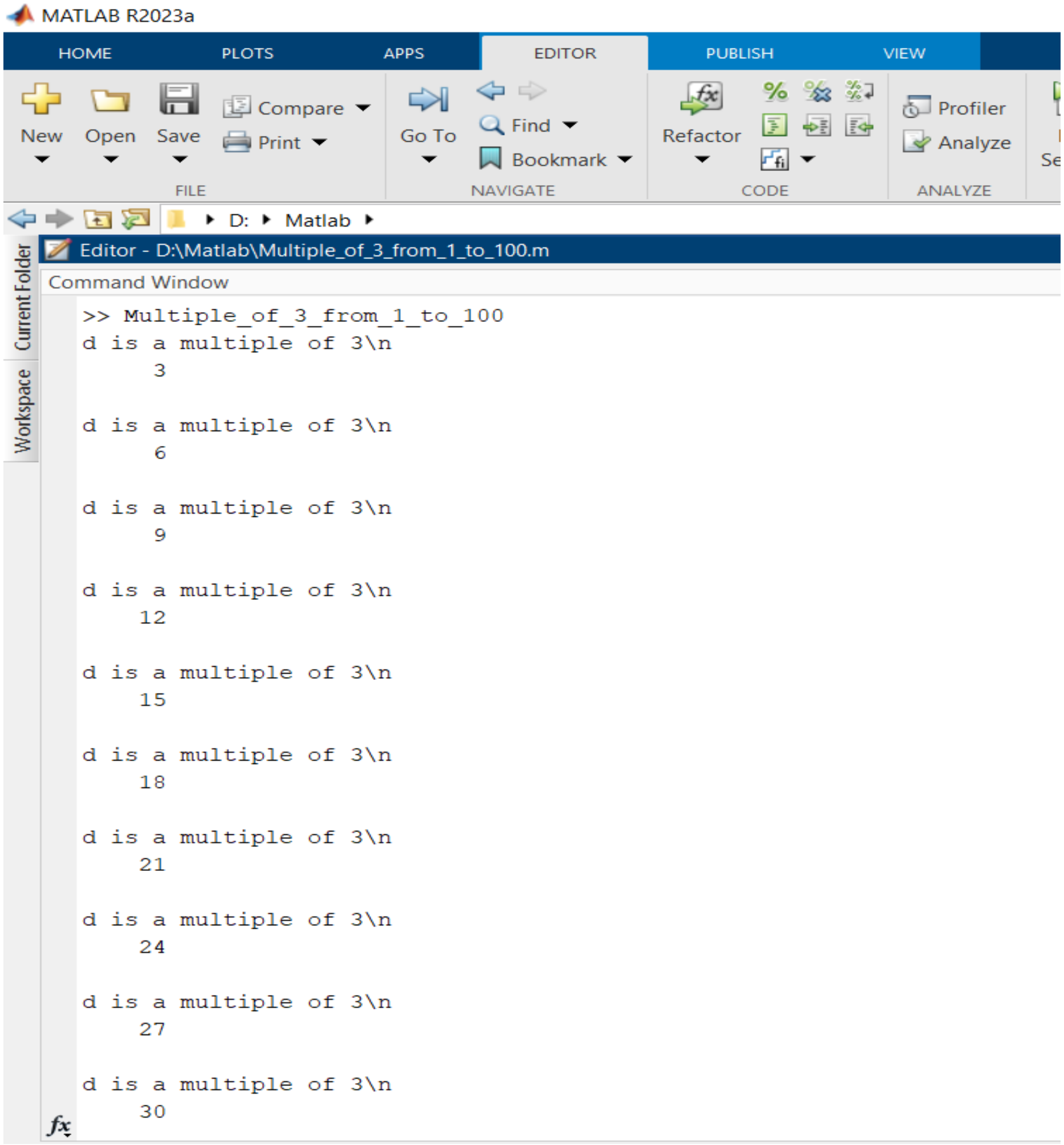
PROGRAM – 6

Aim - WAP in MATLAB to display multiple of 3 between 1 to 100.

SOURCE CODE:

```
% Loop from 1 to 100

for i = 1:100

% Check if the current number is a multiple of 3

if mod(i, 3) == 0

% Display the result

disp('d is a multiple of 3\n');
disp(i)
end
end
```

OUTPUT:



```
>> Multiple_of_3_from_1_to_100
d is a multiple of 3\n
     3

d is a multiple of 3\n
     6

d is a multiple of 3\n
     9

d is a multiple of 3\n
    12

d is a multiple of 3\n
    15

d is a multiple of 3\n
    18

d is a multiple of 3\n
    21

d is a multiple of 3\n
    24

d is a multiple of 3\n
    27

d is a multiple of 3\n
    30
fx
```

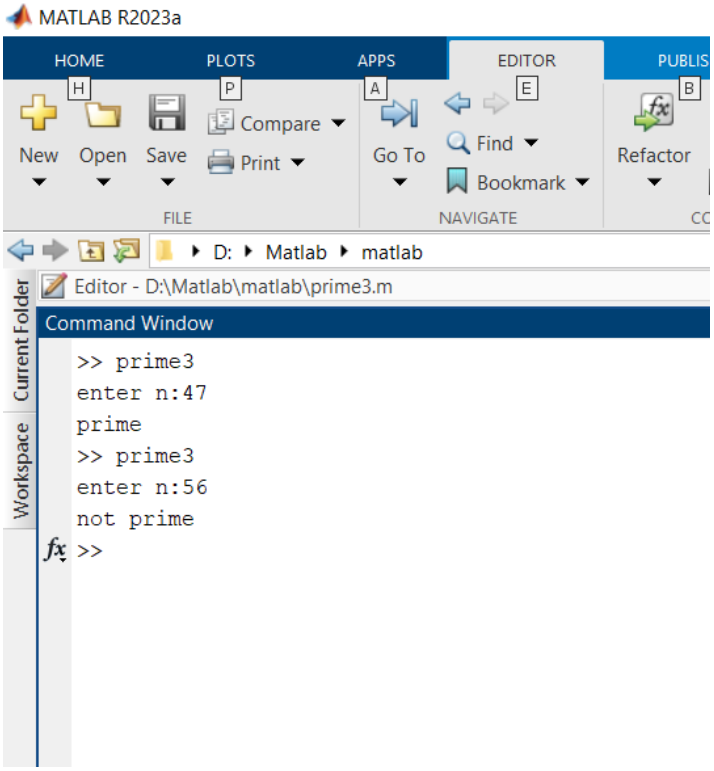
PROGRAM -7

Aim - WAP in MATLAB to check if the number is Prime or Not Prime.

SOURCE CODE:

```
% Enter the number
n = input ('enter n:');

% Initiating a value to a variable
flag = 0;

% Loop check for factors from 2
for i= 2: n/2
r = rem(n,i);
if r==0
flag = 1;
end
end
if flag==1

% Display not prime if n is a factor of 2
disp('not prime')
else

% Display prime if n is not a factor of 2
disp('prime')
end
```

OUTPUT:



```
>> prime3
enter n:47
prime
>> prime3
enter n:56
not prime
>>
```

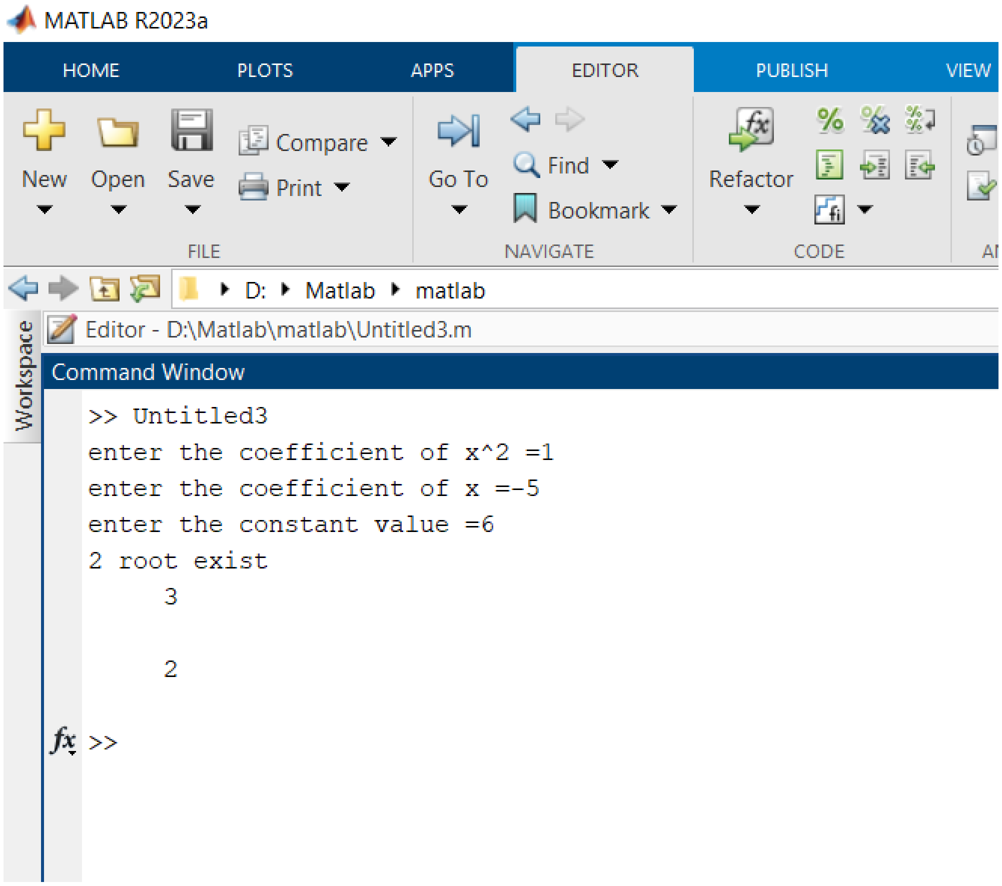
PROGRAM -8

Aim - WAP in MATLAB to find the roots of Quadratic Equation.

SOURCE CODE:

```
% Enter the coefficient of the quadratic equation
a = input('enter the coefficient of x^2 =');
b =  input('enter the coefficient of x =');
c =  input('enter the constant value =');

% Calculate the discriminant
d = sqrt(b^2-4*a*c);

% Check the discriminant to determine the type of roots
if d>0

    % Two real and distinct roots
   disp("2 root exist")
    x1 = (-b+d) ./ (2*a) ;
    disp(x1)
    x2 = (-b-d) ./(2*a) ;
    disp(x2)
elseif d==0

    % One real root (repeated)
    disp("1 root exist")
else

    % No real root present
    disp("no root exist")
end
```

OUTPUT:



```
>> Untitled3
enter the coefficient of x^2 =1
enter the coefficient of x =-5
enter the constant value =6
2 root exist
     3


     2

>>
```

PROGRAM -9

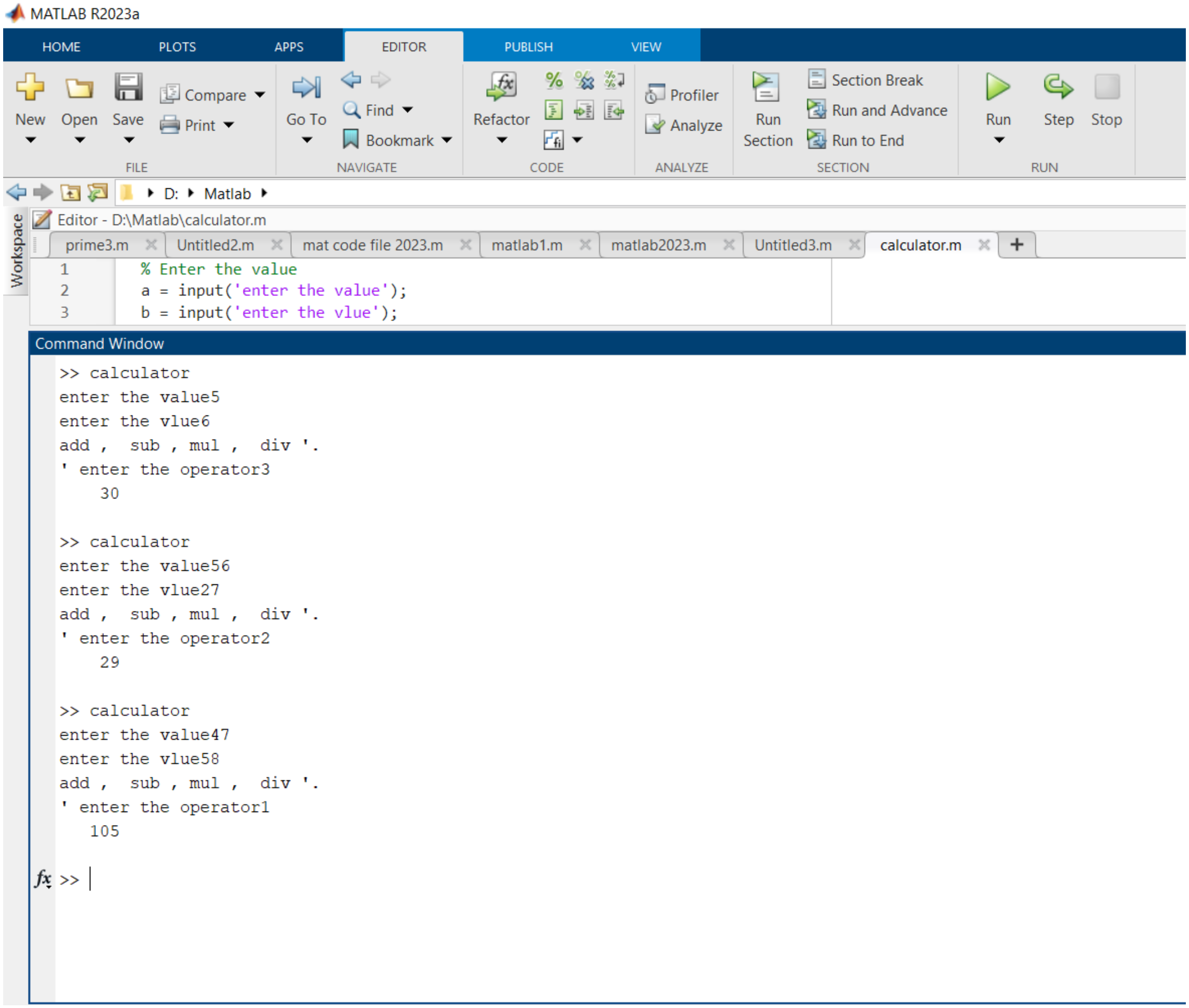
Aim - WAP in MATLAB to make a simple Calculator.

SOURCE CODE:

```
% Enter the value

a = input('enter the value');
b = input('enter the vlue');

% Display different operator of calculator

fprintf( "add ,  sub , mul ,  div '.\n' " )

% Chose the operator

choice = input('enter the operator');
switch choice
    case 1

    % If operator is add

    z = a + b;

    % Display the addition of 2 number

    disp(z)
    case 2

    %If operator is sub

    z = a - b;

    % Display the subtraction of 2 number
```

```
disp(z)
case 3

% If operator is mul

z= a * b;

% Display the multiplication of 2 number

disp(z)
case 4

% If operator is div

z = a / b;

% Display the division of 2 number

disp(z)
end
```

OUTPUT:



```
% Enter the value
a = input('enter the value');
b = input('enter the vlue');
```

```
>> calculator
enter the value5
enter the vlue6
add ,  sub , mul ,  div '.
' enter the operator3
    30

>> calculator
enter the value56
enter the vlue27
add ,  sub , mul ,  div '.
' enter the operator2
    29

>> calculator
enter the value47
enter the vlue58
add ,  sub , mul ,  div '.
' enter the operator1
   105

fx >>
```

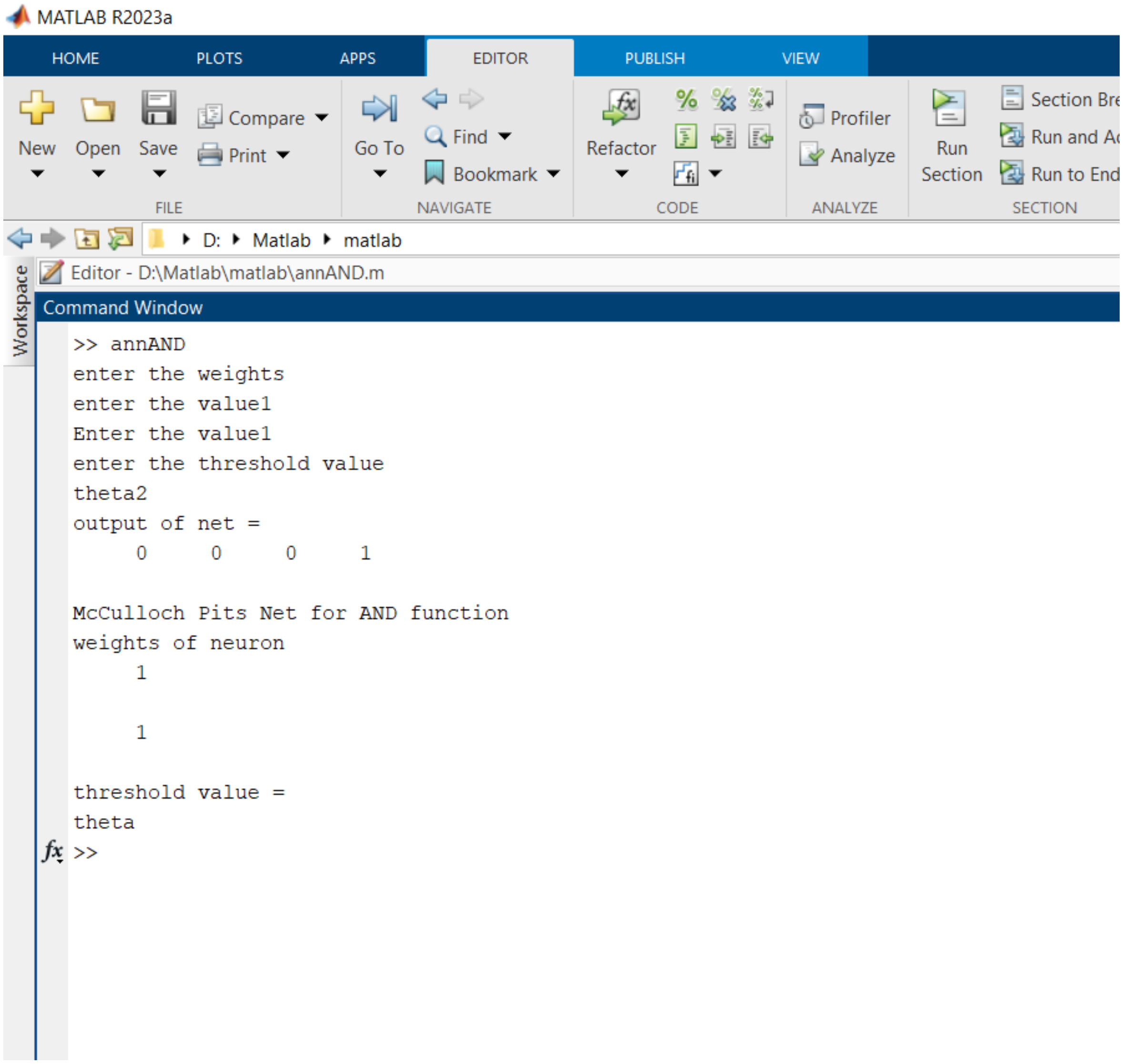
PROGRAM -10

Aim - WAP in MATLAB to implement AND function using McCulloch pits model.

SOURCE CODE:

```
% Getting weight and threshold value
disp('enter the weights')
w1 = input('enter the value');
w2 = input('Enter the value');
disp('enter the threshold value')
theta= input ('theta');

% Define the inputs
y=[0 0 0 0 ];
x1 = [0 1 0 1 ];
x2 = [0 0 1 1 ];
z  = [0 0 0 1 ];

% Compute the output
con = 1 ;
while con
    zin = (x1*w1) + (x2*w2) ;
    for i = 1:4
        if zin(i)>=theta
            y(i) = 1;
        else
            y(i) = 0;
        end
    end

    disp('output of net = ')
    disp(y)
    if y==z
       con = 0;
    else
        disp('Net is not learning enter another set of weights and threshold
value')
    w1 = input('weight w1');
    w2 = input('Weigfht w2');
    theta = input('theta=');
    end
end

% Displaying the values of weights and threshold
disp('McCulloch Pits Net for AND function');
disp('weights of neuron');
disp(w1);
disp(w2)
disp('threshold value =');
disp('theta');
```

OUTPUT:



```
>> annAND
enter the weights
enter the value1
Enter the value1
enter the threshold value
theta2
output of net =
     0      0      0      1

McCulloch Pits Net for AND function
weights of neuron
     1

     1

threshold value =
theta
>>
```

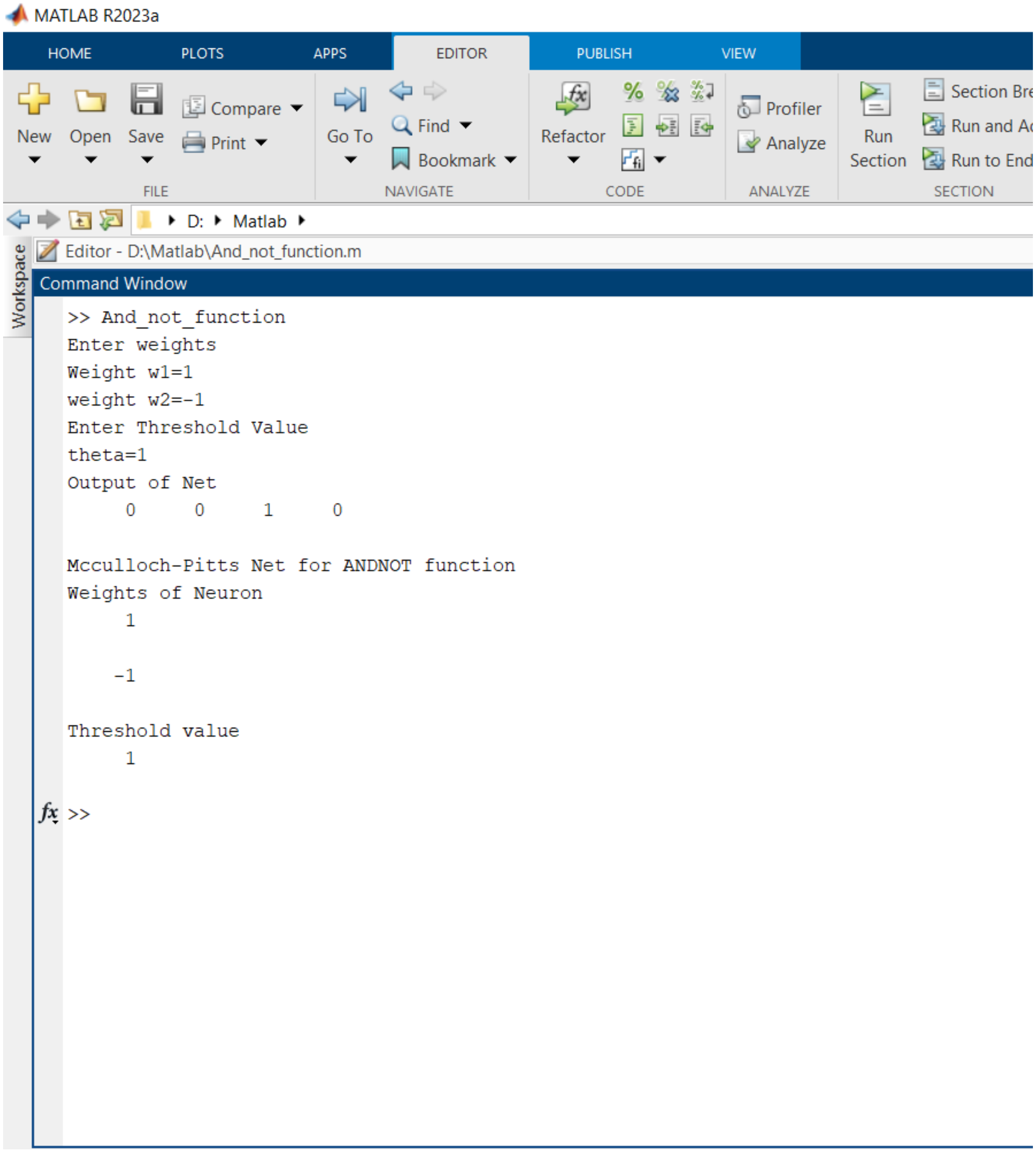
PROGRAM -11

Aim - WAP in MATLAB to implement ANDNOT function using McCulloch pits model.

SOURCE CODE:

```
%ANDNOT function using Mcculloch-Pitts neuron
%Getting weights and threshold value

disp('Enter weights');
w1=input('Weight w1=');
w2=input('weight w2=');
disp('Enter Threshold Value');
theta=input('theta=');

% Define the inputs

y=[0 0 0 0];
x1=[0 0 1 1];
x2=[0 1 0 1];
z=[0 0 1 0];

% Compute the output

con=1;
while con
  zin=x1*w1+x2*w2;
  for i=1:4
      if zin(i)>=theta
          y(i)=1;
      else
          y(i)=0;
      end
  end
  disp('Output of Net');
  disp(y);
  if y==z
      con=0;
  else
      disp('Net is not learning enter another set of weights and Threshold
value');
           w1=input('weight w1=');
           w2=input('weight w2=');
           theta=input('theta=');
  end
end

% Displaying the values of weights and threshol

disp('Mcculloch-Pitts Net for ANDNOT function');
disp('Weights of Neuron');
disp(w1);
disp(w2);
disp('Threshold value');
disp(theta);
```

OUTPUT:



```
>> And_not_function
Enter weights
Weight w1=1
weight w2=-1
Enter Threshold Value
theta=1
Output of Net
     0     0     1     0

Mcculloch-Pitts Net for ANDNOT function
Weights of Neuron
     1

    -1

Threshold value
     1

>>
```

PROGRAM -12

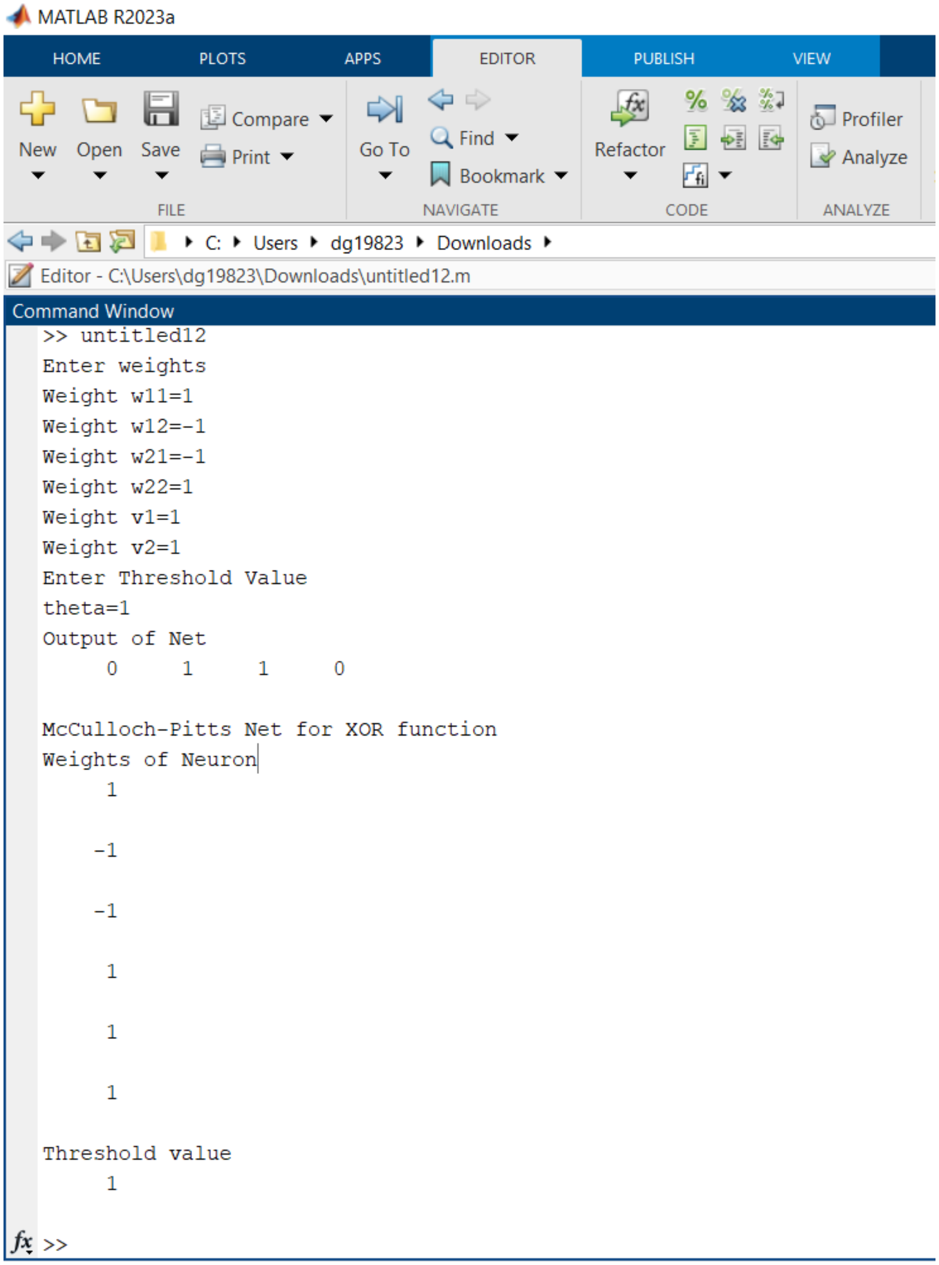
Aim - WAP in MATLAB to implement XOR function using McCulloch pits model.

SOURCE CODE:

```
% Getting Inputs of weights and threshold values
disp('Enter weights');
w11 = input('Weight w11=');
w12 = input('Weight w12=');
w21 = input('Weight w21=');
w22 = input('Weight w22=');
v1 = input('Weight v1=');
v2 = input('Weight v2=');
disp('Enter Threshold Value');
theta = input('theta=');

% Initialize arrays for input and output values
y = [0 0 0 0];
x1 = [0 1 0 1];
x2 = [0 0 1 1];
z = [0 1 1 0];

con = 1;

% Loop for training the network
while con
    zin1 = x1 * w11 + x2 * w21;
    zin2 = x1 * w12 + x2 * w22;

    % Calculate output for the first layer
    for i = 1:4
        if zin1(i) >= theta
            y1(i) = 1;
        else
```

```
            y1(i) = 0;
        end
        if zin2(i) >= theta
            y2(i) = 1;
        else
            y2(i) = 0;
        end
    end

    yin = y1 * v1 + y2 * v2;

    % Calculate final output
    for i = 1:4
        if yin(i) >= theta
            y(i) = 1;
        else
            y(i) = 0;
        end
    end

    disp('Output of Net');
    disp(y);

    % Check if the network has learned the XOR function
    if y == z

       % Terminate the loop if the output matches the expected values
        con = 0;
    else
        disp('Net is not learning. Enter another set of weights and
Threshold value');
        w11 = input('Weight w11=');
        w12 = input('Weight w12=');
        w21 = input('Weight w21=');
        w22 = input('Weight w22=');
        v1 = input('Weight v1=');
        v2 = input('Weight v2=');
        disp('Enter Threshold Value');
        theta = input('theta=');
    end
end

% Display the final weights and threshold value
disp('McCulloch-Pitts Net for XOR function');
disp('Weights of Neuron');
disp(w11);
disp(w12);
disp(w21);
disp(w22);
disp(v1);
disp(v2);
disp('Threshold value');
disp(theta)
```

OUTPUT:



```
>> untitled12
Enter weights
Weight w11=1
Weight w12=-1
Weight w21=-1
Weight w22=1
Weight v1=1
Weight v2=1
Enter Threshold Value
theta=1
Output of Net
     0     1     1     0

McCulloch-Pitts Net for XOR function
Weights of Neuron
     1

    -1

    -1

     1

     1

     1

Threshold value
     1

>>
```

PROGRAM -13

Aim - WAP in MATLAB to Plot various Membership Functions.

SOURCE CODE:

```
% Illustration of various membership functions
% Define the range of values for the x-axis

x=(0.0:1.0:10.0);

% Triangular membership function

y1=trimf(x,[1 3 5]);

% Plot the membership functions

subplot(3,1,1)
plot(x,y1);
title('Triangular Membership Function');
xlabel('x');
ylabel('Membership');

% Define the range of values for the x-axis

x=(0.0:1.0:10.0);

% Trapezoidal membership function
y2=trapmf(x,[1 3 5 7]);

% Plot the membership functions

subplot(3,1,2)
```

```
plot(x,y2 , 'g');
title('Trapezoidal Membership Function');
xlabel('x');
ylabel('Membership');

% Define the range of values for the x-axis

x=(0.0:2.0:10.0);

% Bell shaped membership function

y3=gbellmf(x,[1 2 5]);

% Plot the membership functions

subplot(3,1,3)
plot(y3 , 'r');
title(' Membership Function');
xlabel('x');
ylabel('Membership');
```

OUTPUT:

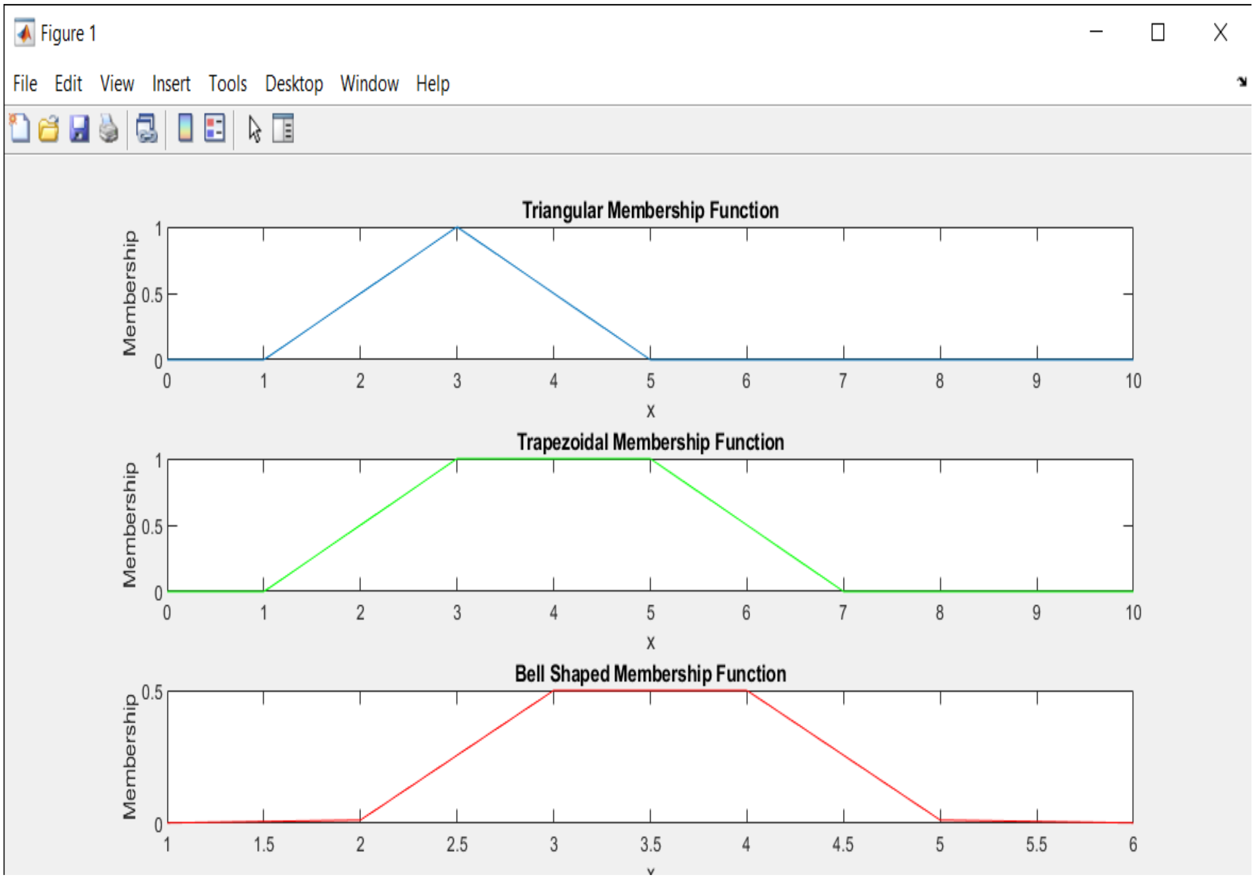


Figure 1
File Edit View Insert Tools Desktop Window Help
Triangular Membership Function
Membership
1
0.5
0
0 1 2 3 4 5 6 7 8 9 10
x
Trapezoidal Membership Function
Membership
1
0.5
0
0 1 2 3 4 5 6 7 8 9 10
x
Bell Shaped Membership Function
Membership
0.5
0
1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6
x

PROGRAM -14

Aim - WAP in MATLAB to generate a few activation functions that are being used in neural network

SOURCE CODE:

```
% Activation Functions in Neural Networks
% Define the range of x values
x = -10:0.1:10;
tmp = exp(-x);

%Sigmoid Activation Function

y1 = 1./(1 + tmp);

%Hyperbolic Activation Function

y2 = (1-tmp)./(1+tmp);

% Linear Activation Function

y3 = x;

% Plot the Activation Functions figure
% Plotting of Logistic Activation Function

subplot(231);
plot(x,y1);
grid on;
axis([min(x)   max(x) -2 2]);
title ('logistic function');
xlabel('a');
axis('square');
```

```
% Plotting of Hyperbolic Activation Function

subplot(232);
plot(x,y2)
axis([min(x)  max(x) -2  2 ])
grid on
title('hyperbolic function')
xlabel('b')
axis ('square')

% Plotting of Linear Activation Function

subplot(233)
plot(x , y3)
xlabel('c')
grid on
axis([min(x)  max(x) -10 10])

axis("square")
title('linear function')
```

OUTPUT:

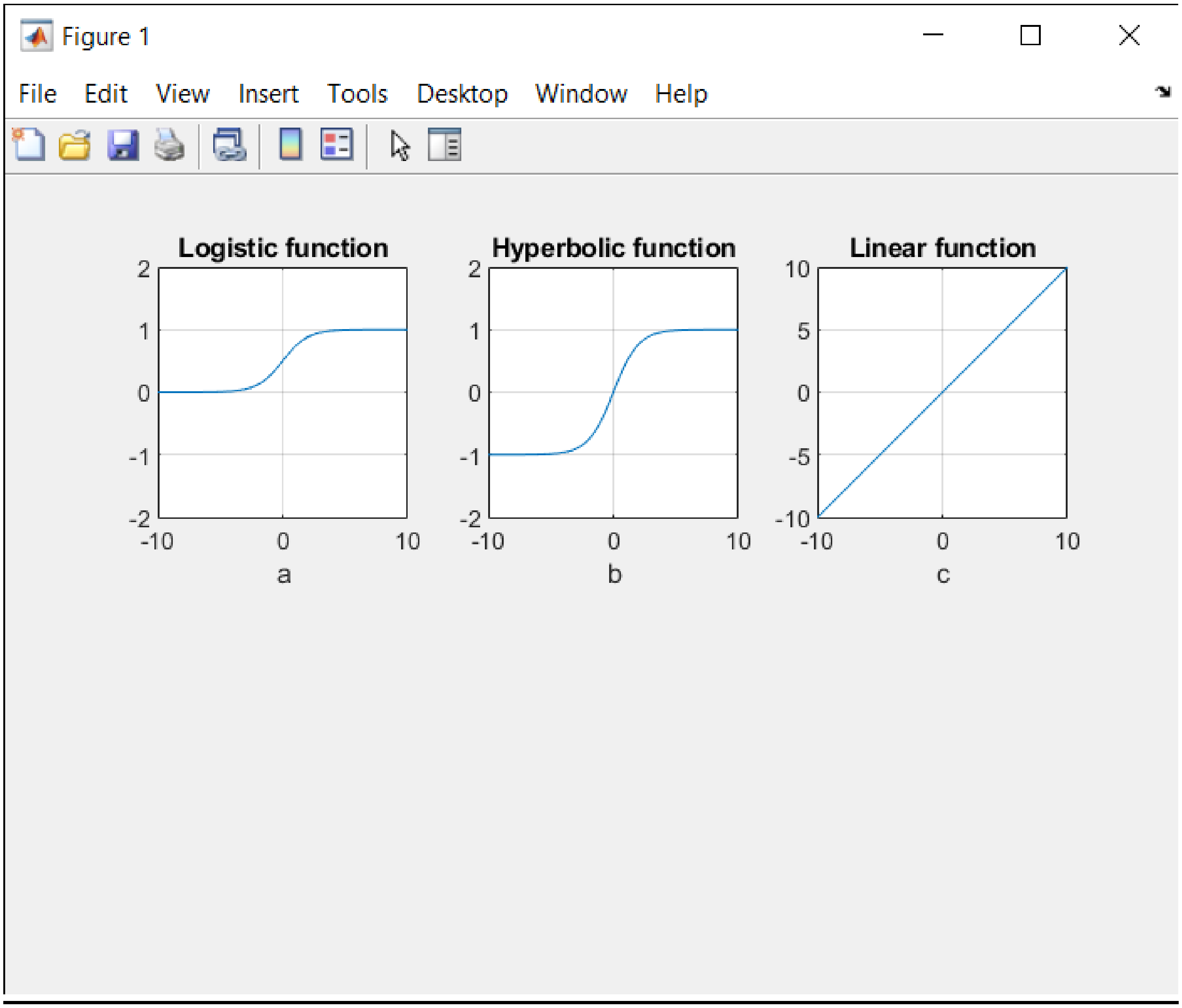


Figure 1
File Edit View Insert Tools Desktop Window Help
Logistic function
Hyperbolic function
Linear function
2
1
0
-1
-2
-10
0
10
a
b
c
10
5
-5
-10

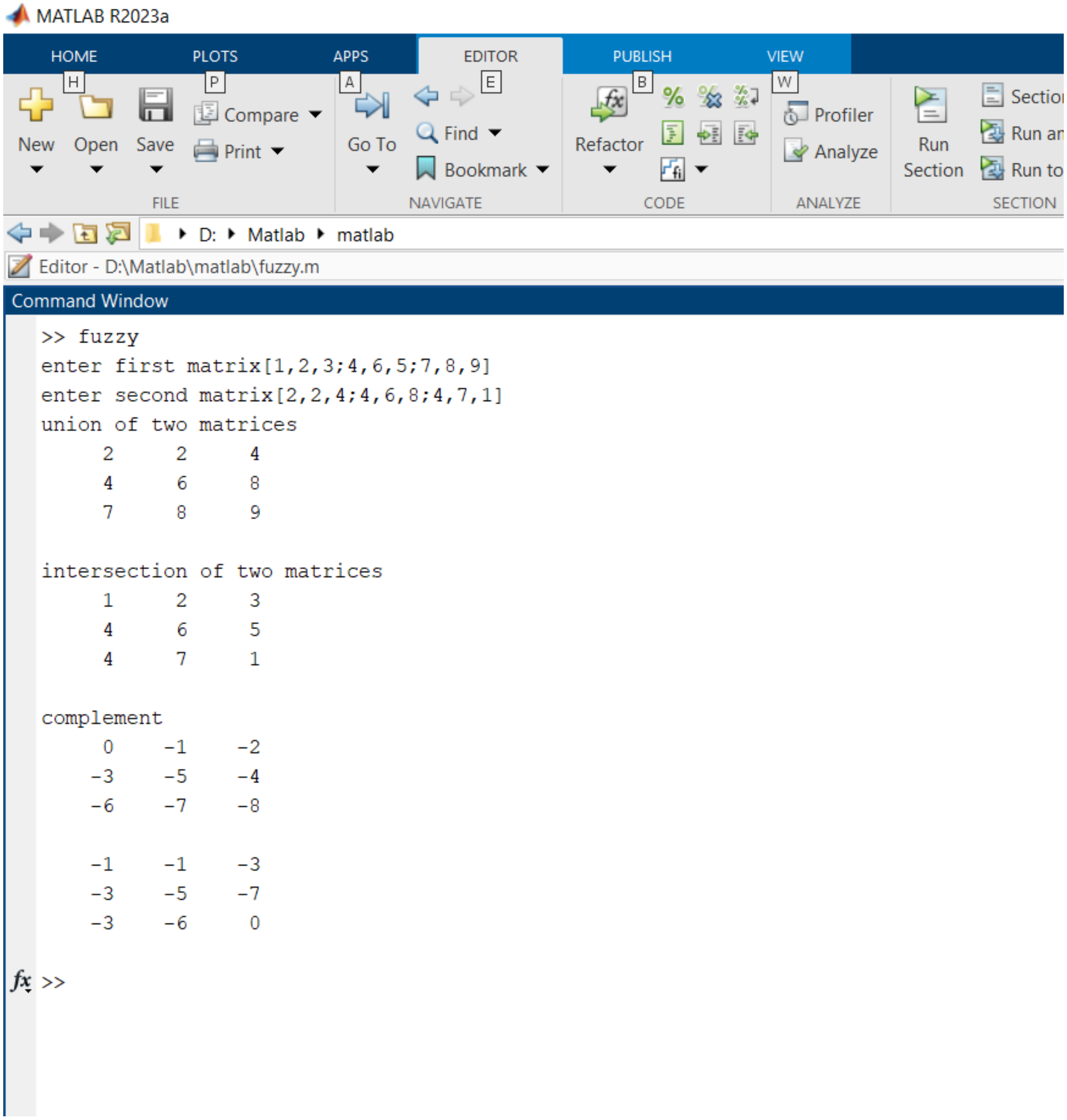
PROGRAM -15

Aim - WAP in MATLAB to perform Fuzzy set Operations.

SOURCE CODE:

```
% Enter the matrix values
u=input('enter first matrix');
v=input('enter second matrix');
% Union of matrix
w=max(u,v);
% Intersection of matrix
p=min(u,v);
% Complement of the matrix
q1=1-u;
q2=1-v;
% Display of union matrix
disp('union of two matrices');
disp(w);
% Display of intersection of matrix
disp('intersection of two matrices');
disp(p);
% Display of complement of u and v matrix
disp('complement');
disp(q1);
disp(q2);
```

OUTPUT:



```
>> fuzzy
enter first matrix[1,2,3;4,6,5;7,8,9]
enter second matrix[2,2,4;4,6,8;4,7,1]
union of two matrices
     2     2     4
     4     6     8
     7     8     9

intersection of two matrices
     1     2     3
     4     6     5
     4     7     1

complement
     0    -1    -2
    -3    -5    -4
    -6    -7    -8

    -1    -1    -3
    -3    -5    -7
    -3    -6     0

>>
```

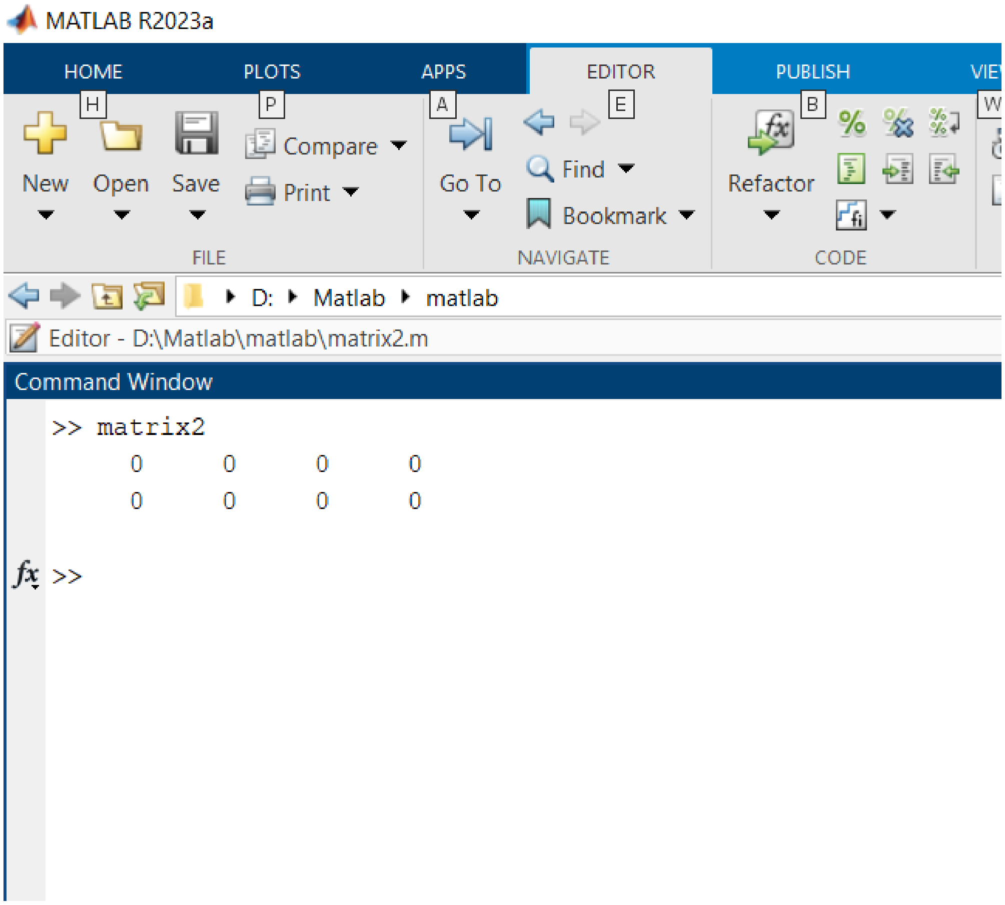
PROGRAM -16

Aim - WAP in MATLAB to create

(a) Create a matrix of zeros with 2 rows and 4 columns

SOURCE CODE (a):

```
% Create a matrix of zeros with 2 rows and 4 columns
matrix = zeros(2, 4);

% Display the
```



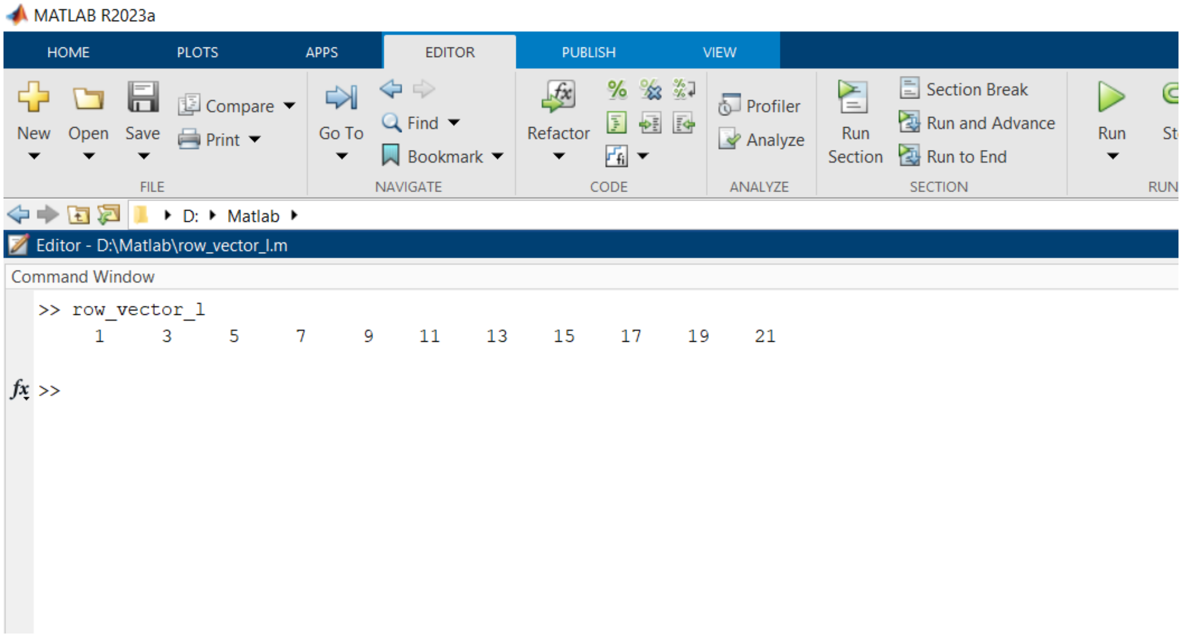
```
>> matrix2
     0     0     0     0
     0     0     0     0

>>
```

(b) Create the row vector of odd numbers through 21, L = 1 3 5 7 9 11 13 15 17 19 21 Use the colon operator.

SOURCE CODE(b):

```
% Create a row vector of odd numbers from 1 to 21

L = 1:2:21;

% Display the resulting vector

disp(L);
```

OUTPUT:



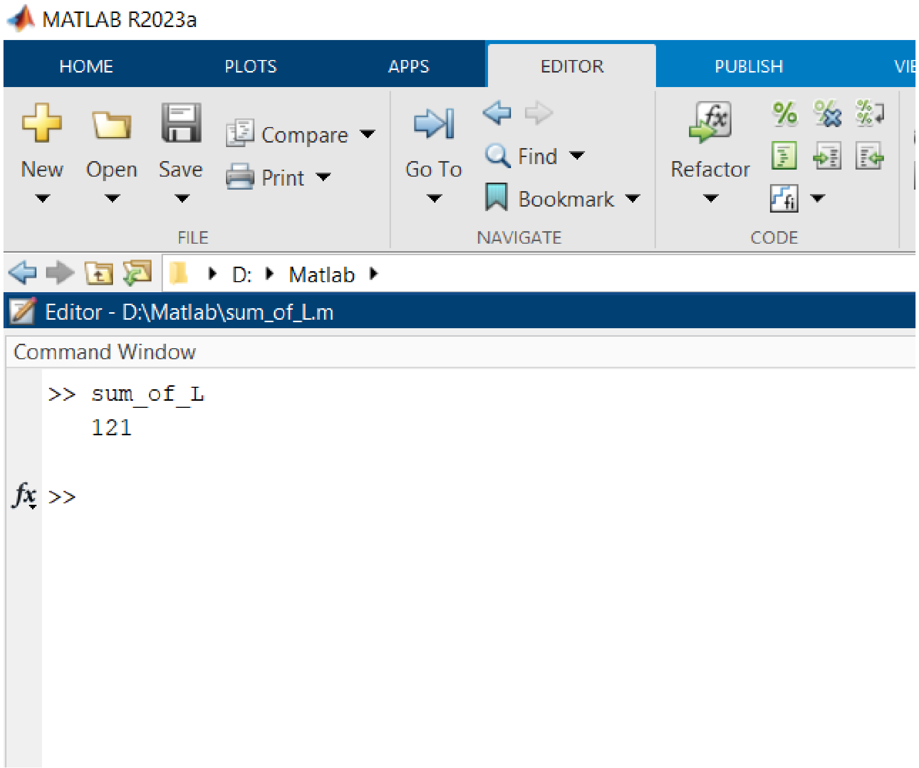
```
>> row_vector_l
     1     3     5     7     9    11    13    15    17    19    21

>>
```

(c) Find the sum S of vector L's elements.

SOURCE CODE(c):

```
% Create the row vector of odd numbers from 1 to 21
L = 1:2:21;

% Calculate the sum of the elements in vector L
S = sum(L);

% Display the sum
disp(S);
```

OUTPUT:



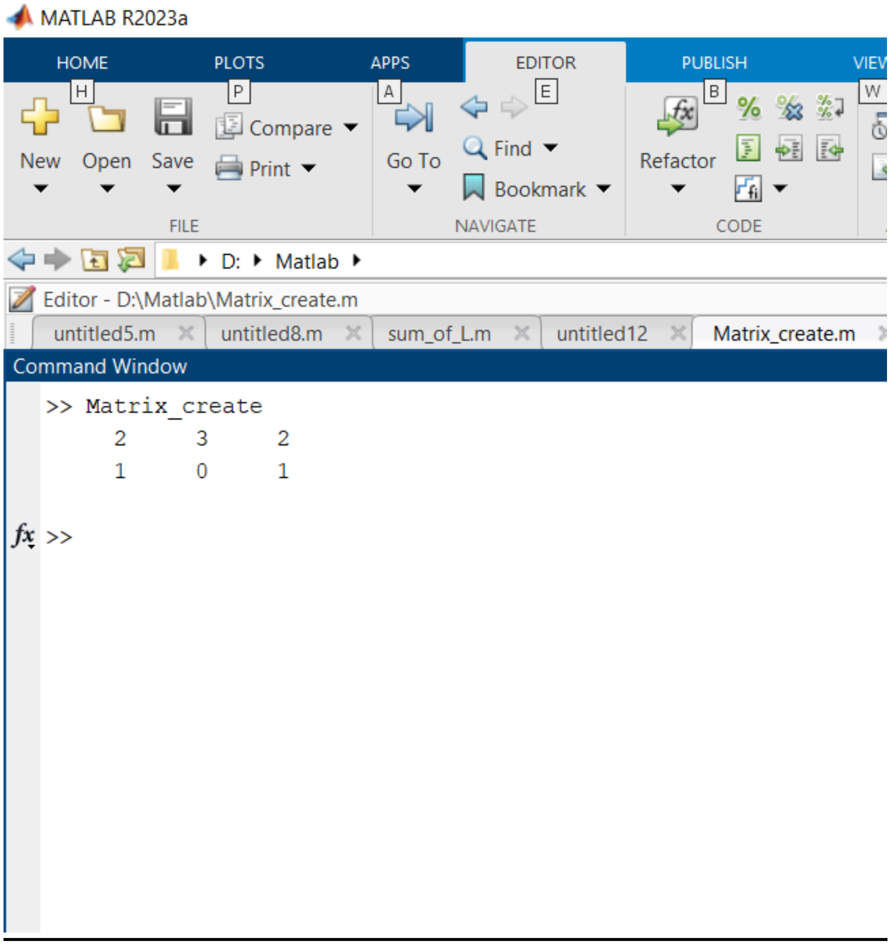
```
>> sum_of_L
   121

>>
```

(d) Form the matrix A = 2 3 2 1 0 1.

SOURCE CODE(d):

```
% Define the matrix A
A = [2, 3, 2; 1, 0, 1];

% Display the matrix A
disp(A);
```

OUTPUT:



```
>> Matrix_create
     2     3     2
     1     0     1

>>
```

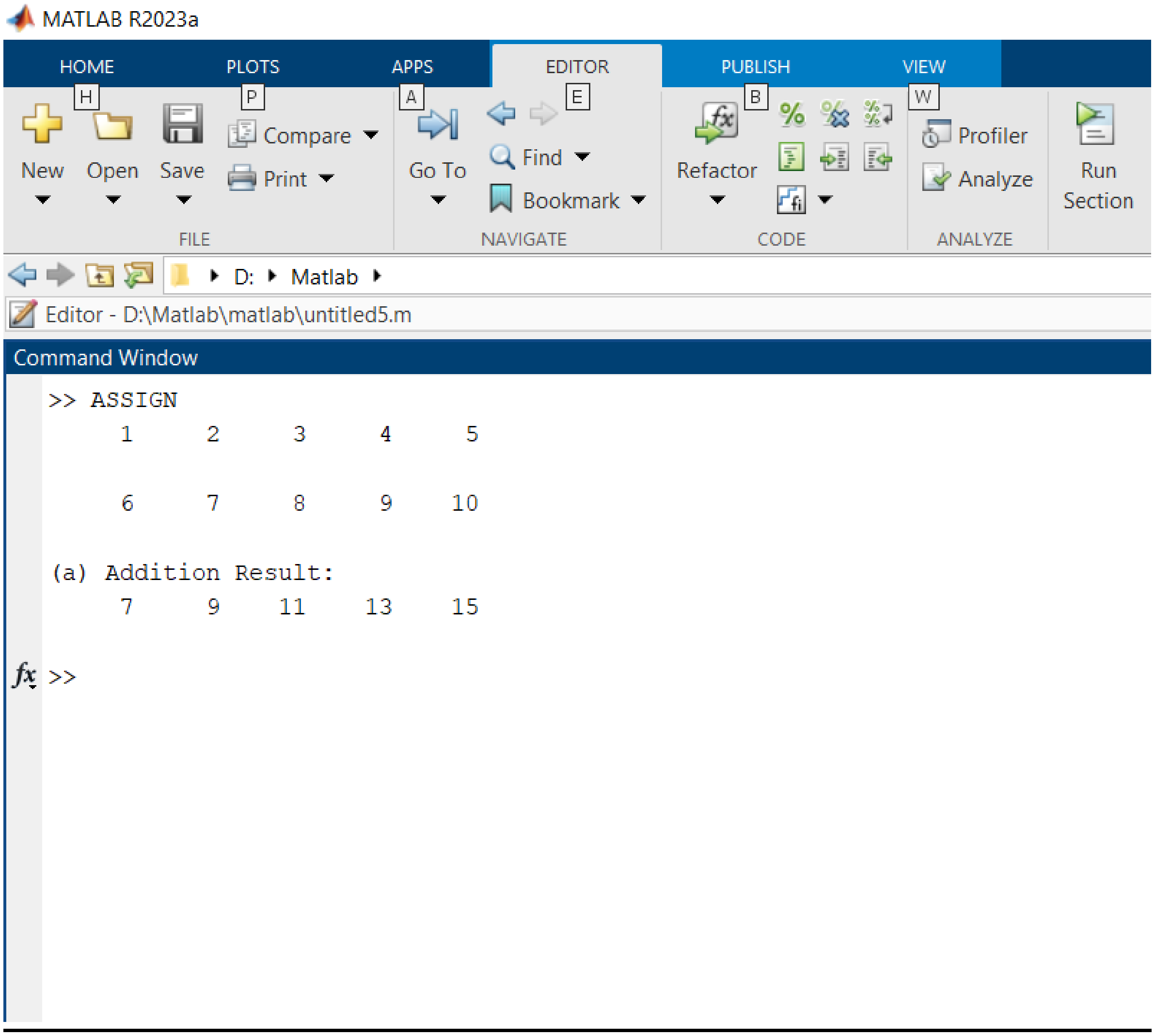
PROGRAM -17

Aim - WAP in MATLAB to create
(a) Create two different vectors of the same length and add them.

SOURCE CODE(a):

```
% Vectors of the same length and add them.

vector1 = [1, 2, 3, 4, 5];

vector2 = [6, 7, 8, 9, 10];

add = vector1 + vector2;

% Display the vector

disp(vector1)

disp(vector2)

% Display of addition result

disp('(a) Addition Result:');

disp(add);
```

OUTPUT:



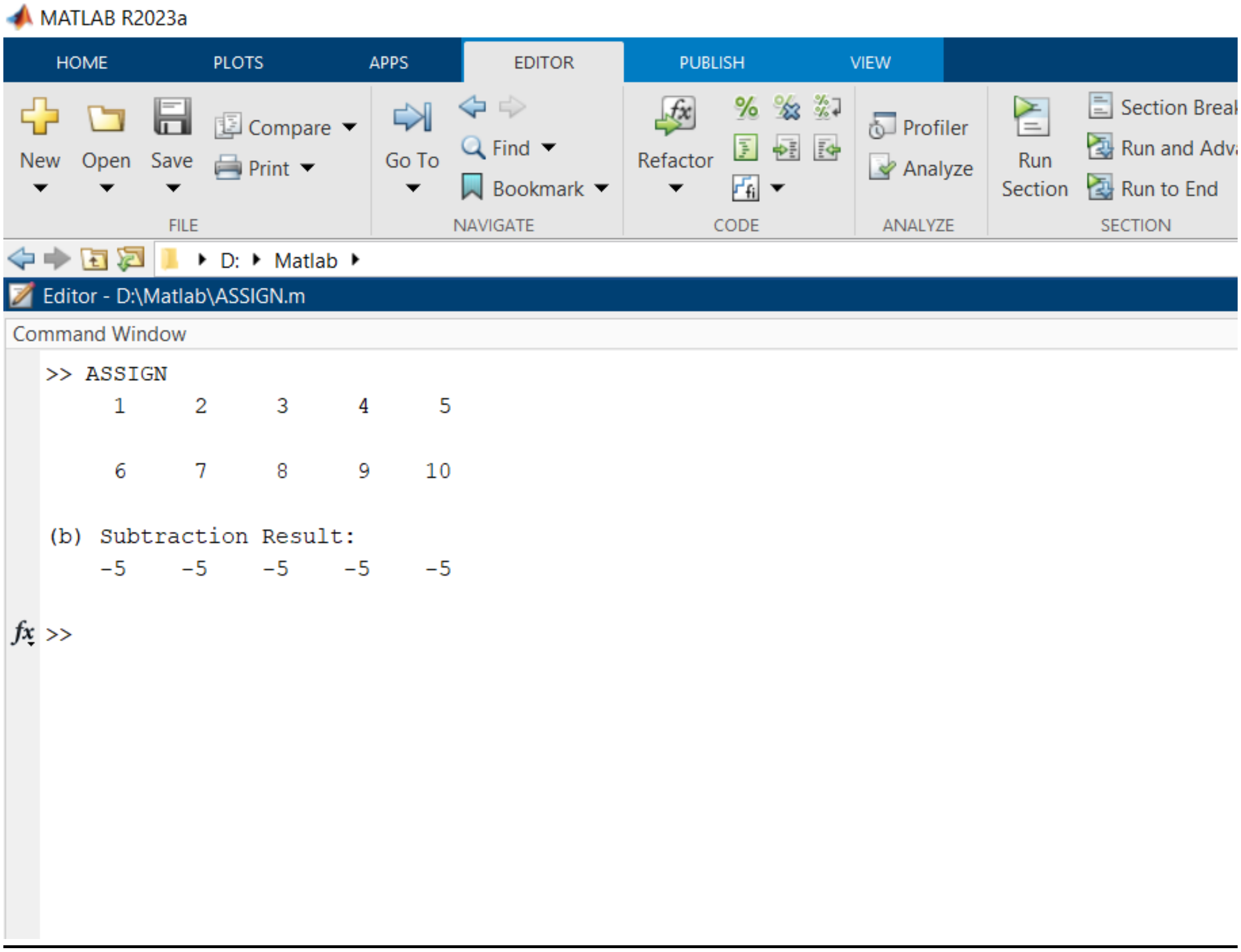
```
>> ASSIGN
     1     2     3     4     5


     6     7     8     9    10

(a) Addition Result:
     7     9    11    13    15

>>
```

(b) Now subtract them.

SOURCE CODE(b):

```
% Vectors of the same length and add them.
vector1 = [1, 2, 3, 4, 5];
vector2 = [6, 7, 8, 9, 10];
% Display the vector
disp(vector1)
disp(vector2)
% Now subtract them.
sub = vector1 - vector2;
% Display the subtraction result
disp('(b) Subtraction Result:');
disp(sub);
```

OUTPUT:



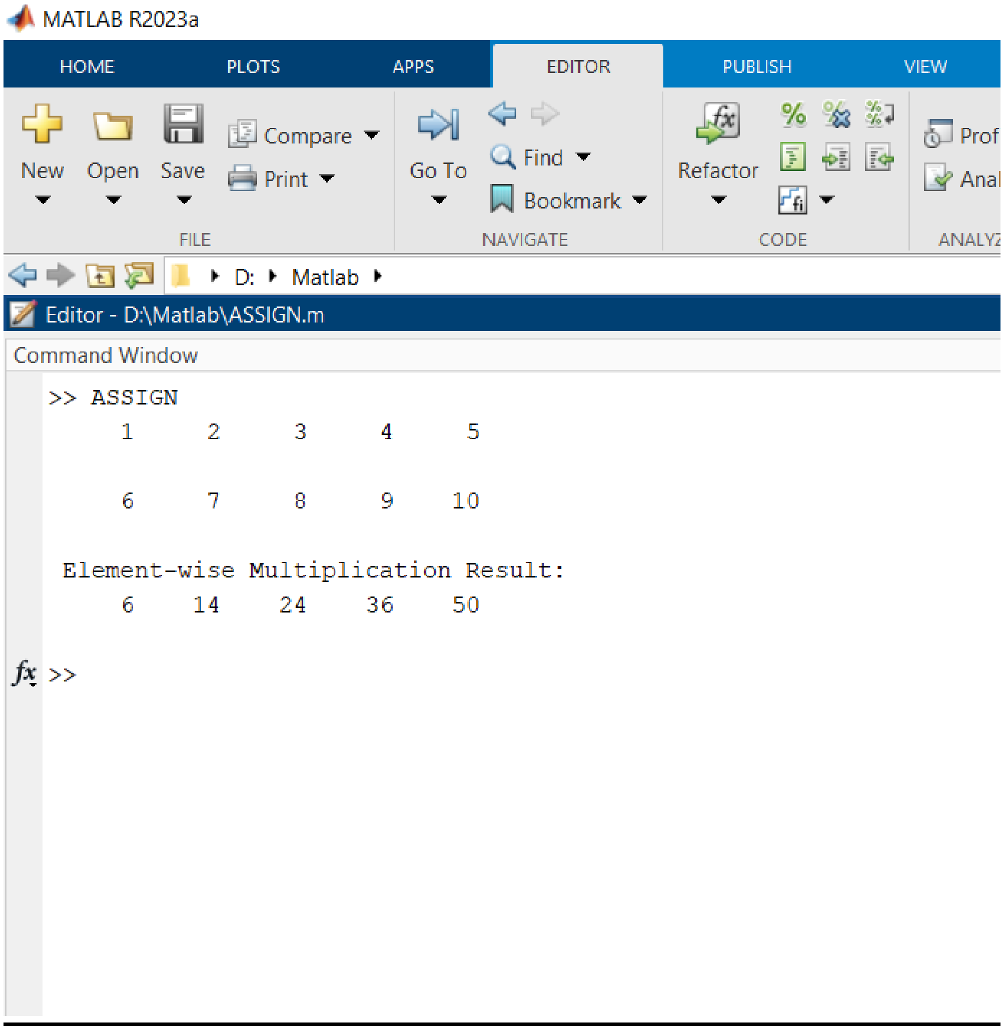
```
>> ASSIGN
     1     2     3     4     5


     6     7     8     9    10

(b) Subtraction Result:
    -5    -5    -5    -5    -5

>>
```

(c) Perform element-by-element multiplication on them.

SOURCE CODE(c):

```
% Vectors of the same length and add them.

vector1 = [1, 2, 3, 4, 5];

vector2 = [6, 7, 8, 9, 10];

% Display the vector

disp(vector1)

disp(vector2)

%  Perform element-by-element multiplication on them.

mul = vector1 .* vector2;

% Display Multiplication Result

disp(' Element-wise Multiplication Result:');

disp(mul);
```

OUTPUT:



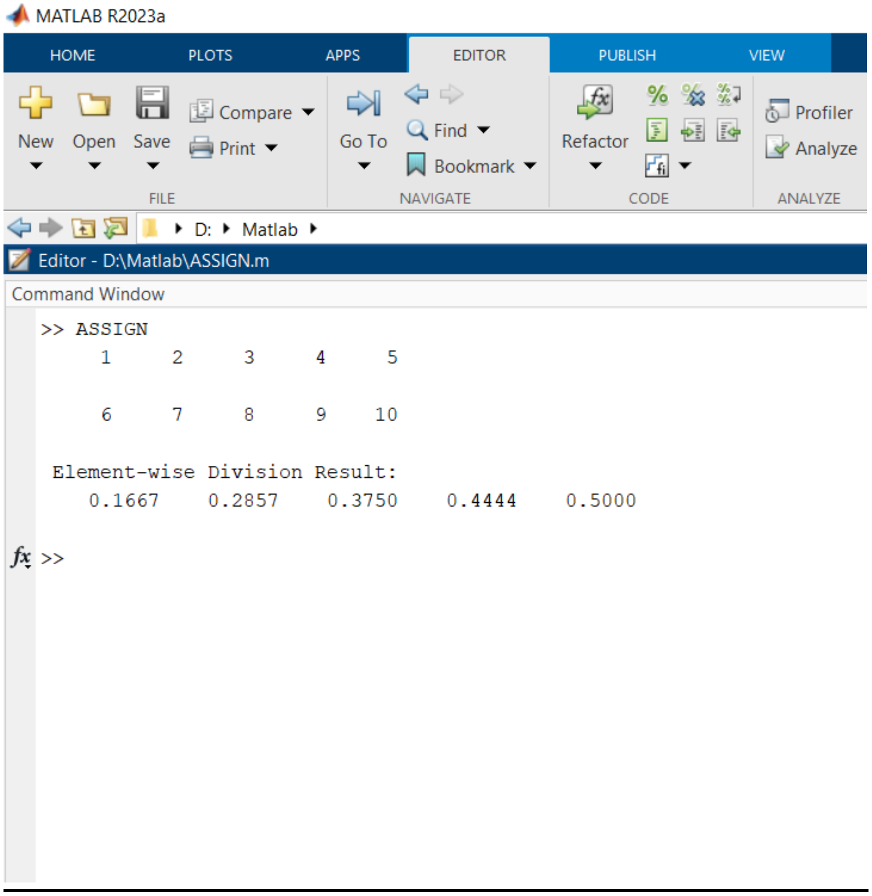
```
>> ASSIGN
     1     2     3     4     5


     6     7     8     9    10

 Element-wise Multiplication Result:
     6    14    24    36    50

>>
```

(d) Perform element-by-element division on them.

SOURCE CODE(d):

```
% Vectors of the same length and add them.

vector1 = [1, 2, 3, 4, 5];

vector2 = [6, 7, 8, 9, 10];

% Display the vector

disp(vector1)

disp(vector2)

%  Perform element-by-element division on them.

div = vector1 ./ vector2;

% Display the Element-wise Division Result

disp(' Element-wise Division Result:');

disp(div);
```

OUTPUT:



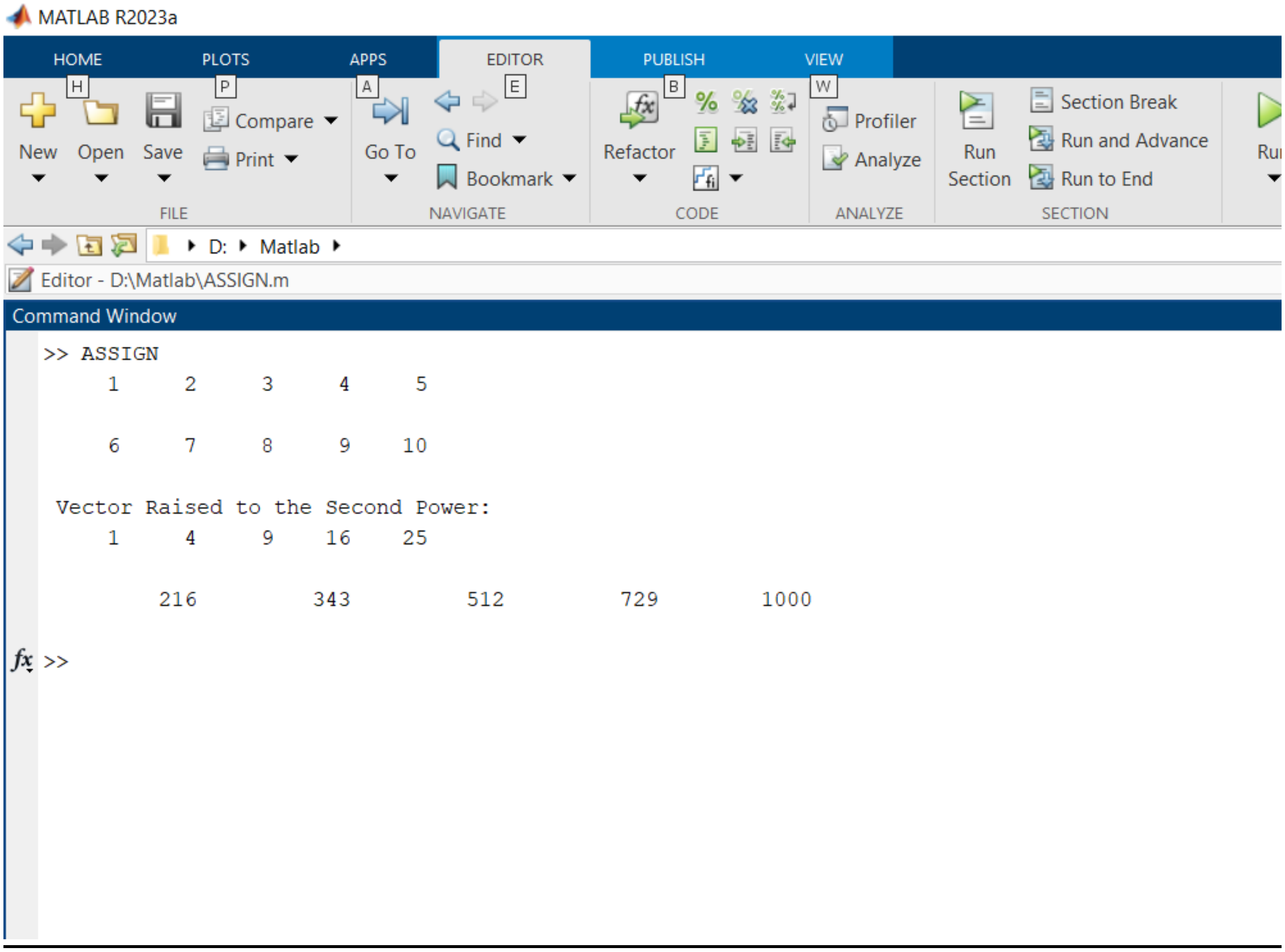
```
>> ASSIGN
     1     2     3     4     5

     6     7     8     9    10

 Element-wise Division Result:
    0.1667    0.2857    0.3750    0.4444    0.5000

>>
```

(e)Raise one of the vectors to the second power.

SOURCE CODE(e):

```
% Vectors of the same length and add them.
vector1 = [1, 2, 3, 4, 5];
vector2 = [6, 7, 8, 9, 10];
% Display the vector
disp(vector1)
disp(vector2)
%  Raise one of the vectors to the second power.
a = vector1.^2;
b = vector2.^3;
% Display the result
disp(' Vector Raised to the Second Power:');
disp(a);
disp(b);
```

OUTPUT:



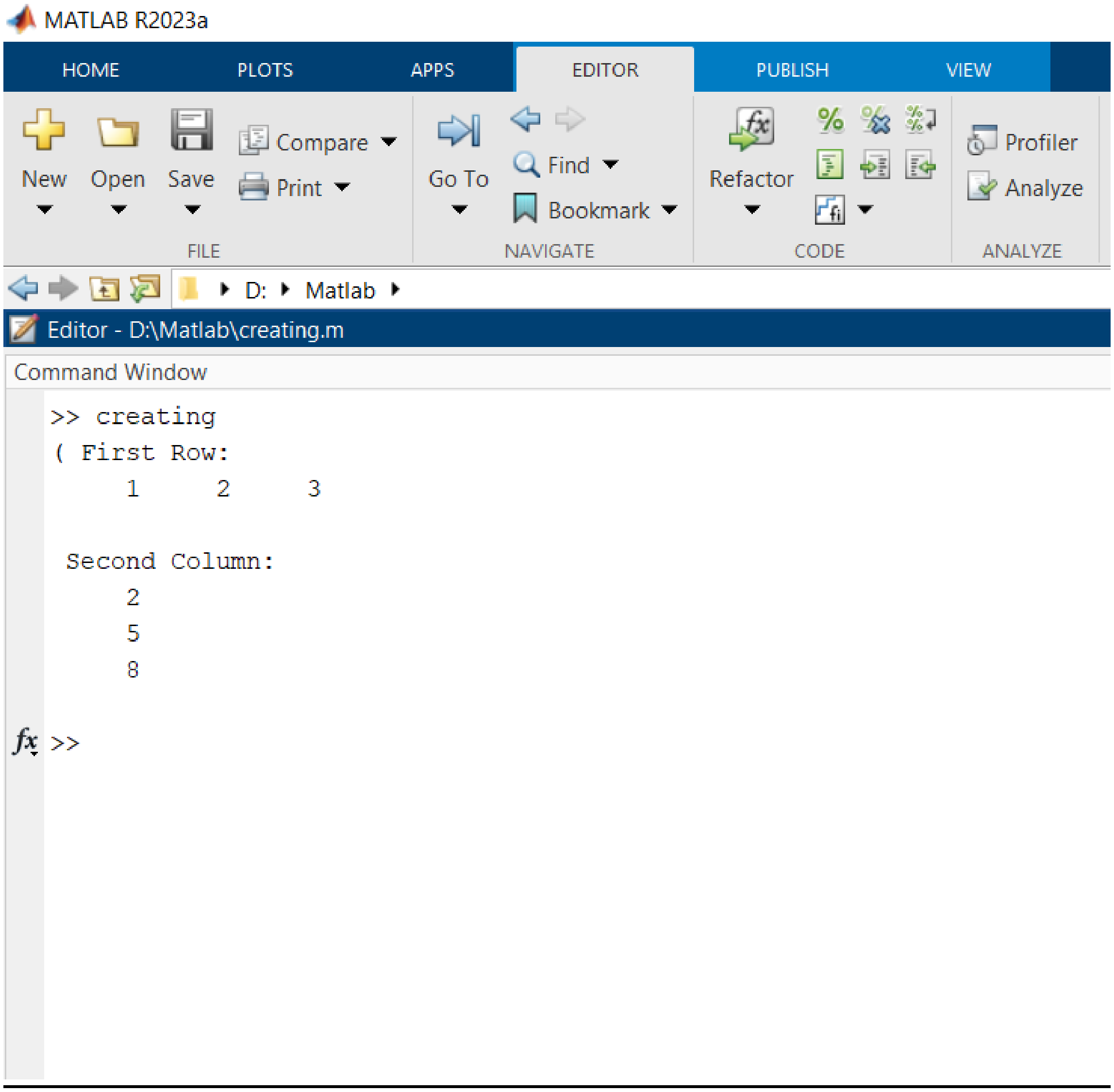
```
>> ASSIGN
     1     2     3     4     5


     6     7     8     9    10

 Vector Raised to the Second Power:
     1     4     9    16    25


         216         343         512         729        1000

>>
```

(f) Create a 3 × 3 matrix and display the first row of and the second column on the screen.

SOURCE CODE(f):

```
% Create a 3x3 matrix and display the first row and the second column.

matrix = [1, 2, 3; 4, 5, 6; 7, 8, 9];

% Display first row

disp('First Row:');

disp(matrix(1, :));

% Display second column

disp(' Second Column:');

disp(matrix(:, 2));
```

OUTPUT:



```
>> creating
( First Row:
     1      2      3

 Second Column:
     2
     5
     8

>>
```

PROGRAM -18

Aim - WAP in MATLAB Using the plot command for multiple plots, plot y = sin(x) and y = cos(x) on the same graph for values of x defined by: x = 0: pi/30:2*pi.

SOURCE CODE:

```
% Define the range of x values
x = 0:pi/30:2*pi;

% Calculate the corresponding y values for sin(x) and cos(x)
y1 = sin(x);
y2 = cos(x);

% Plot sin(x) in the subplot
subplot(2,1,1);
plot(x,y1 , 'b')
xlabel('X')
title('Plot of sin(x)');

% Plot cos(x) in the subplot
subplot(2,1,2);
plot(x,y2 , 'r')
xlabel('Y')
title('Plot of cos(x)');

% Add a title to the entire figure
sgtitle('Plot of sin(x) and cos(x)');
```

OUTPUT:

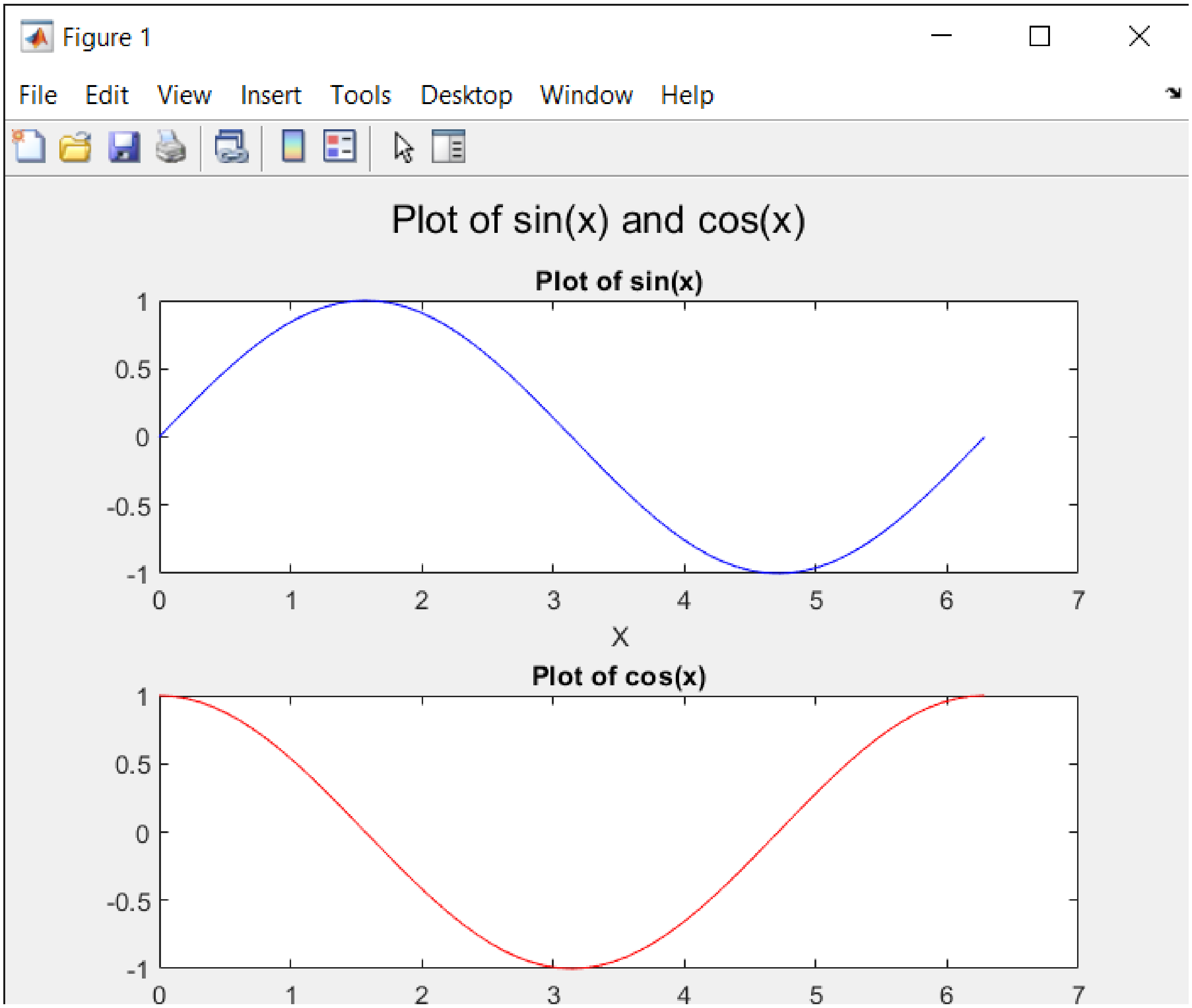


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Plot of sin(x) and cos(x)
Plot of sin(x)
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
X
Plot of cos(x)
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7

PROGRAM -19

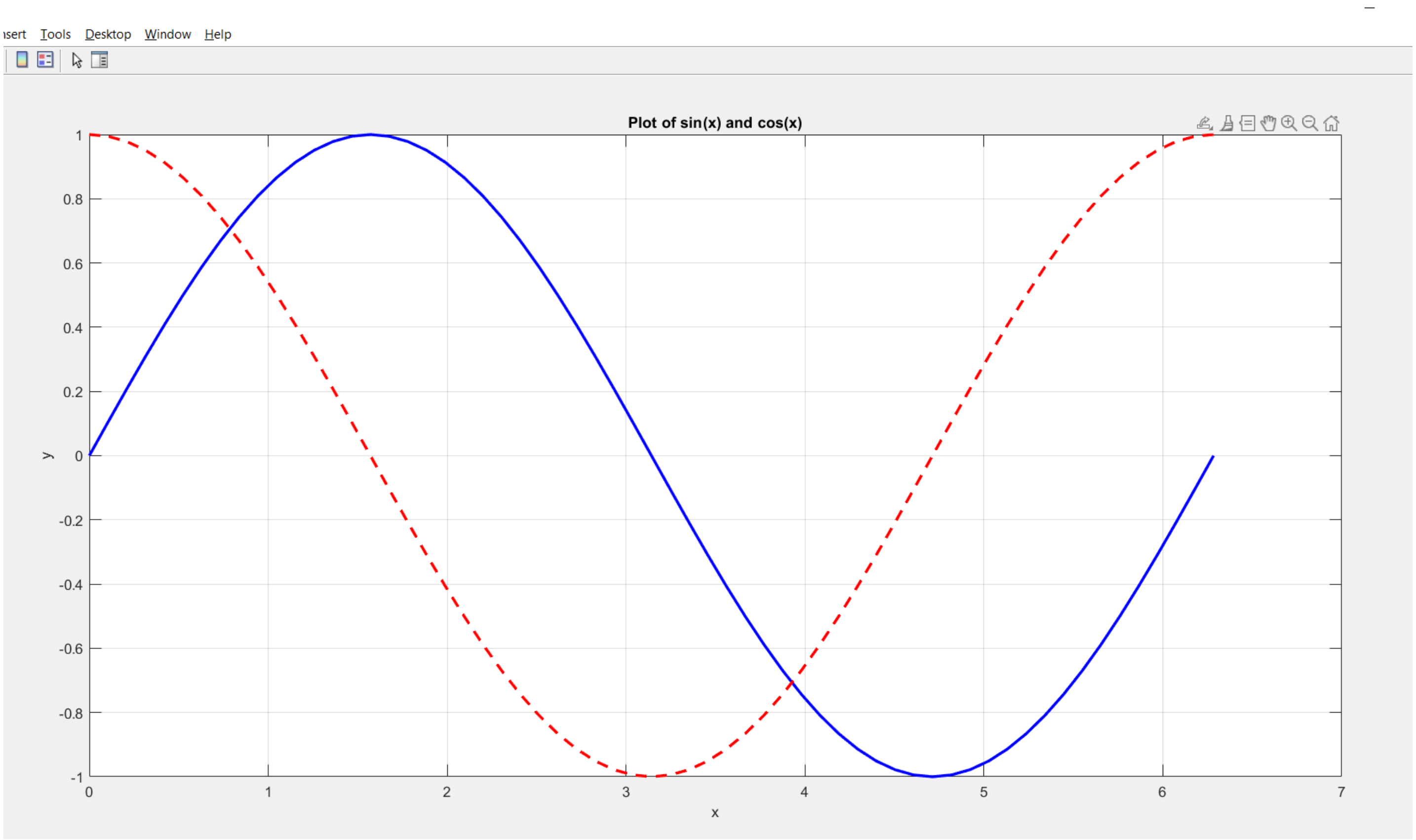
Aim - WAP in MATLAB Using the plot command for a single plot and the hold commands, plot y = sin(x) and y = cos(x) on the same graph for values of x defined by: x = 0: pi/30:2*pi.

SOURCE CODE:

```
% Define the range of x values

x = 0:pi/30:2*pi;

% Calculate the corresponding y values for sin(x) and cos(x)

y1 = sin(x);

y2 = cos(x);

% Create the plotfigure;

% Plot sin(x) in blue

plot(x, y1, 'b-', 'LineWidth', 2,'DisplayName', 'sin(x)');

hold on;

% Hold the current plot

% Plot cos(x) in red dashed

plot(x, y2, 'r--','LineWidth’, 2,'DisplayName', 'cos(x)');

% Release the current plot
```

```
hold off;

% Add labels and legend

xlabel('x');

ylabel('y');

title('Plot of sin(x) and cos(x)');

% Add grid lines (optional)

grid on;
```

OUTPUT:



nsert
Tools
Desktop
Window
Help
Plot of sin(x) and cos(x)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y
0
1
2
3
4
5
6
7
x

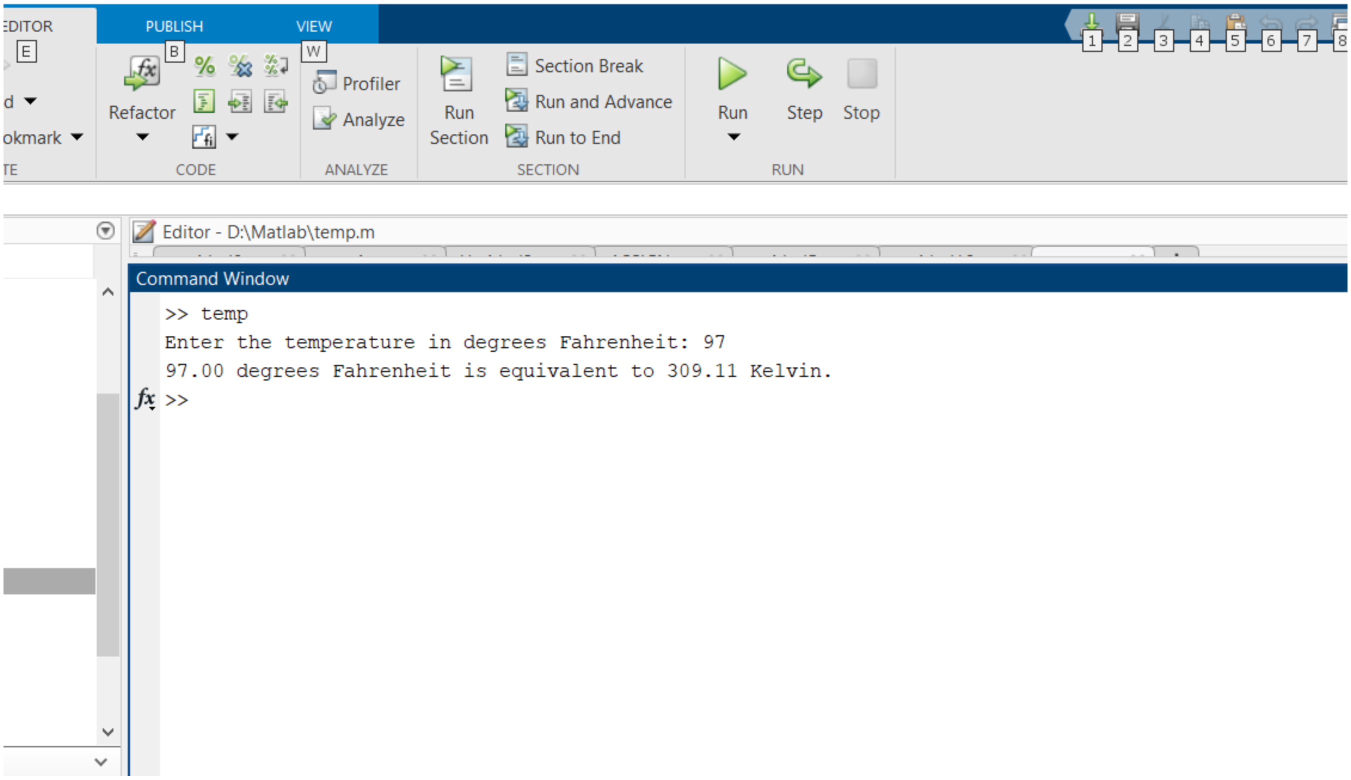
PROGRAM -20

Aim - (a) write a program in MATLAB that reads an input temperature in degree Fahrenheit, converts it to an absolute temperature in kelvin. Hint: TK = [5/9 (TF-32) +273], TF = 97;

SOURCE CODE(a):

```
% Prompt the user for input temperature in Fahrenheit

TF = input('Enter the temperature in degrees Fahrenheit: ');

% Convert Fahrenheit to Kelvin using the formula

TK = (5/9) * (TF - 32) + 273;

% Display the result

fprintf('%.2f degrees Fahrenheit is equivalent to %.2f Kelvin.\n', TF, TK);
```

OUTPUT:



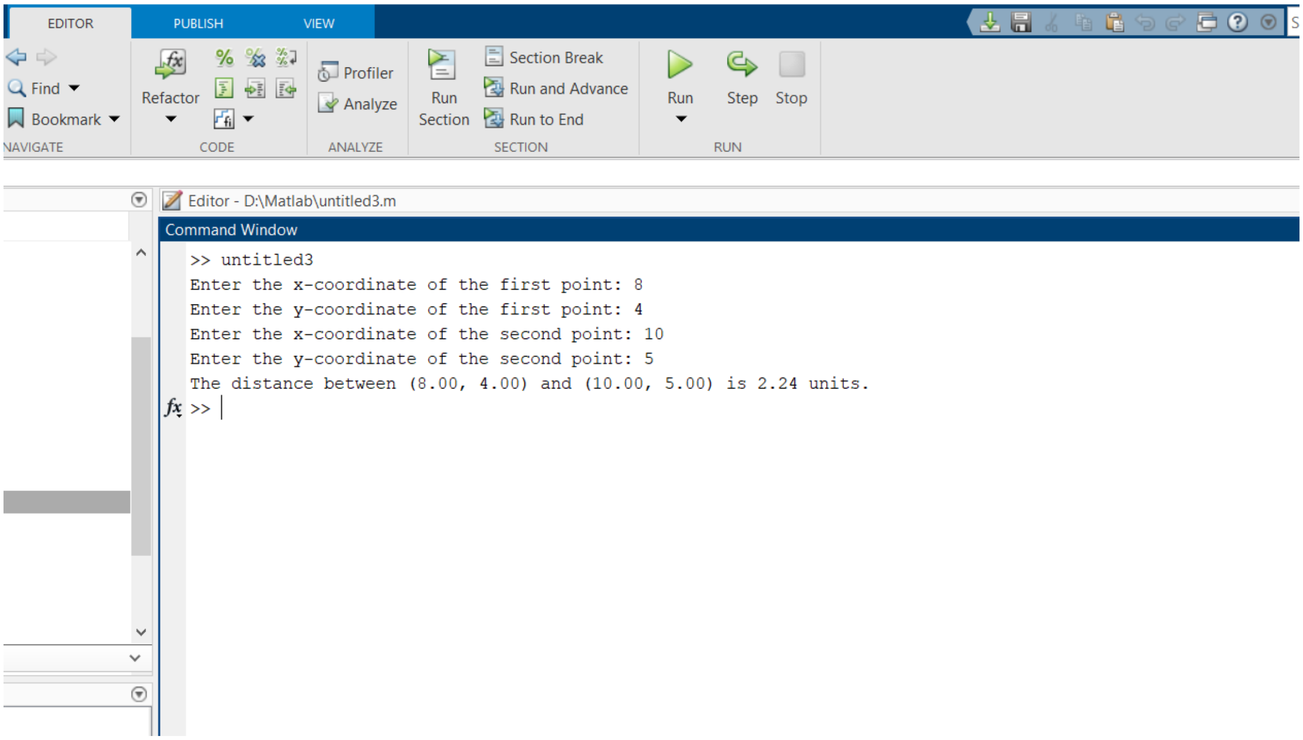
```
>> temp
Enter the temperature in degrees Fahrenheit: 97
97.00 degrees Fahrenheit is equivalent to 309.11 Kelvin.
>>
```

Aim - (b) Write a program to find out the distance between two points (x1, y1) and (x2,y2) specified by the cartesian coordinate plane. Hint: d = sqrt $(x1\text{-}x2)^2 + (y1\text{-}y2)^2$.

SOURCE CODE:

```
% Prompt the user for input coordinates

x1 = input('Enter the x-coordinate of the first point: ');

y1 = input('Enter the y-coordinate of the first point: ');

x2 = input('Enter the x-coordinate of the second point: ');

y2 = input('Enter the y-coordinate of the second point: ');

% Calculate the distance using the distance formula

d = sqrt((x1 - x2)^2 + (y1 - y2)^2);

% Display the result

fprintf('The distance between (%.2f, %.2f) and (%.2f, %.2f) is %.2f units.\n', x1, y1, x2, y2, d);
```

OUTPUT:



```
>> untitled3
Enter the x-coordinate of the first point: 8
Enter the y-coordinate of the first point: 4
Enter the x-coordinate of the second point: 10
Enter the y-coordinate of the second point: 5
The distance between (8.00, 4.00) and (10.00, 5.00) is 2.24 units.
>> 
```

PROGRAM -21

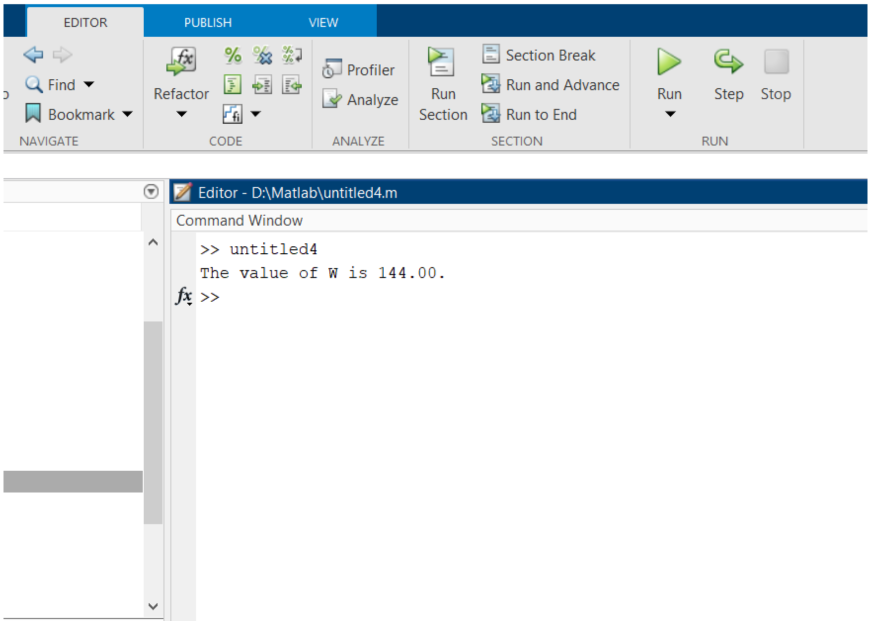
Aim - Suppose that x=3 and y=4. Use the MATLAB to evaluate the following expression:

(a) $W = x^2y^2 / (x-y)^2$

SOURCE CODE:

```
% Define the values of x and y
x = 3;
y = 4;

% Calculate the expression
W = (x^2 * y^2) / ((x - y)^2);

% Display the result
fprintf('The value of W is %.2f.\n', W);
```

OUTPUT:



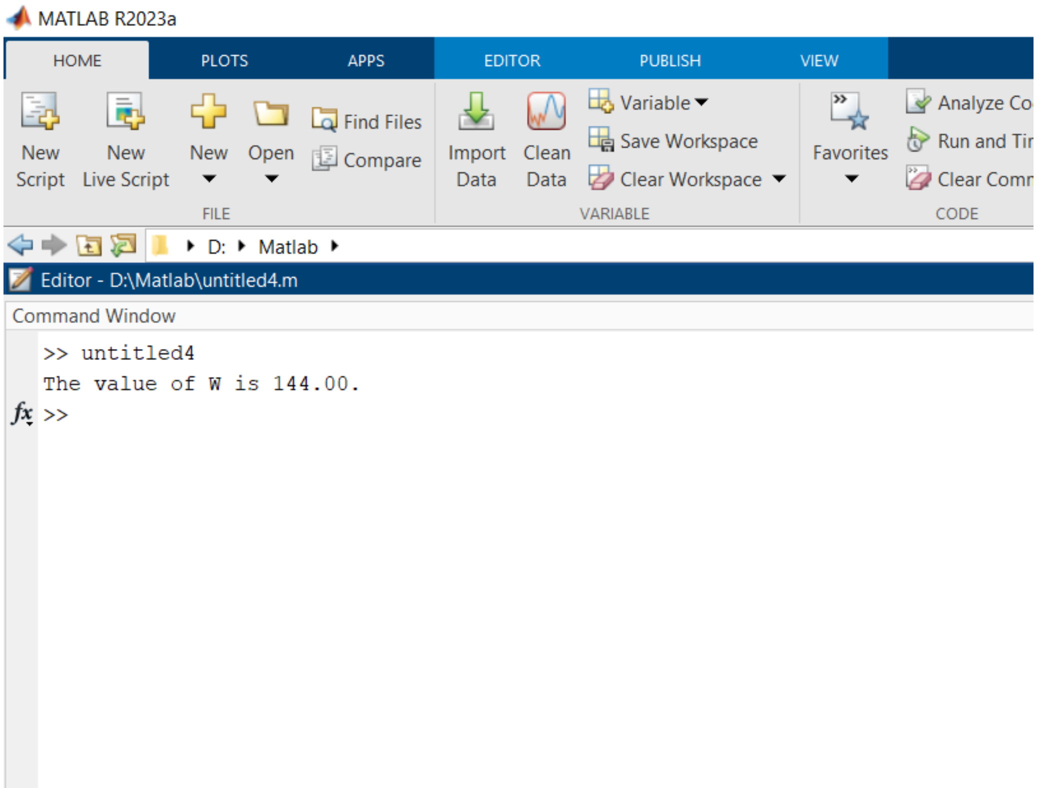
```
>> untitled4
The value of W is 144.00.
fx >>
```

(b) $Z = 2 / y^2(x+y)^2$

<u>SOURCE CODE:</u>

```
% Define the values of x and y
x = 3;
y = 4;

% Calculate the expression
Z = 2 / (y.^2)*(x+y).^2;

% Display the result
fprintf('The value of W is %.2f.\n', W);
```

<u>OUTPUT:</u>



```
>> untitled4
The value of W is 144.00.
>>
```

PROGRAM- 22

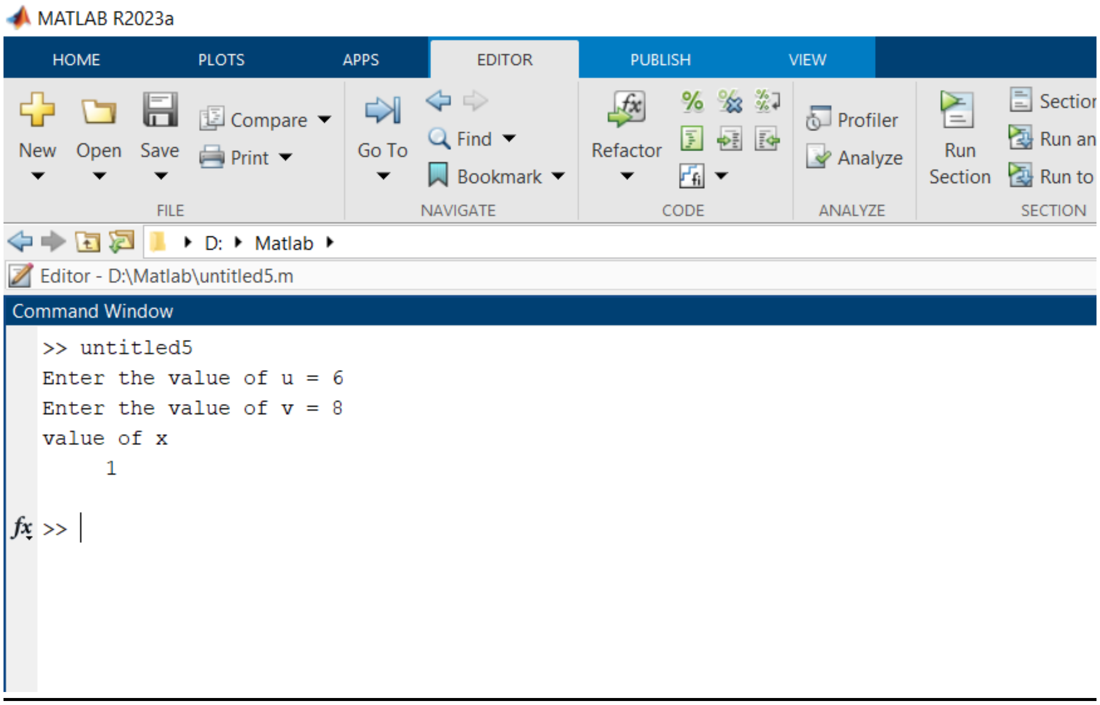
Aim - Write the following expression in MATLAB

(1) x = 4u / 3v

SOURCE CODE:

```
% Define the values of u and v
u = input('Enter the value of u');
v = input('Enter the value of v');

% Calculate the expression
x = (4*u) ./ (3*v) ;

% Display the result
disp('value of x');
disp(x)
```

OUTPUT:



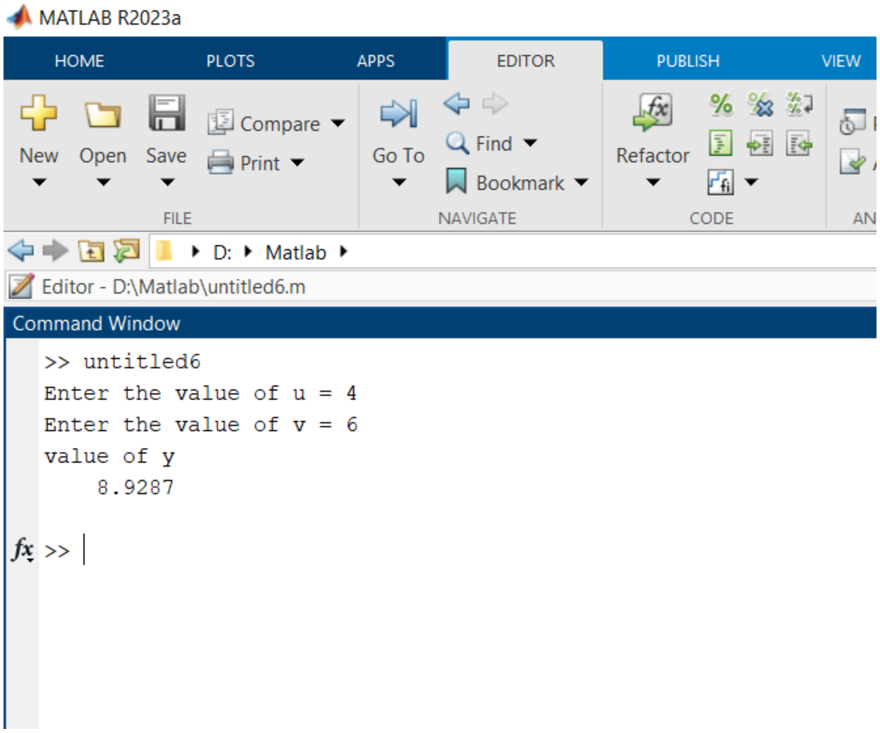
```
>> untitled5
Enter the value of u = 6
Enter the value of v = 8
value of x
     1

>> |
```

(2) $y = v^3\, 2\pi / v^3 - u^3$

SOURCE CODE:

```
% Define the values of u and v
u = input('Enter the value of u = ');
v = input('Enter the value of v = ');

% Calculate the expression
y = v.^3 *2*pi ./ (v.^3 - u.^3);

% Display the result
disp('value of y ');
disp(y)
```

OUTPUT:



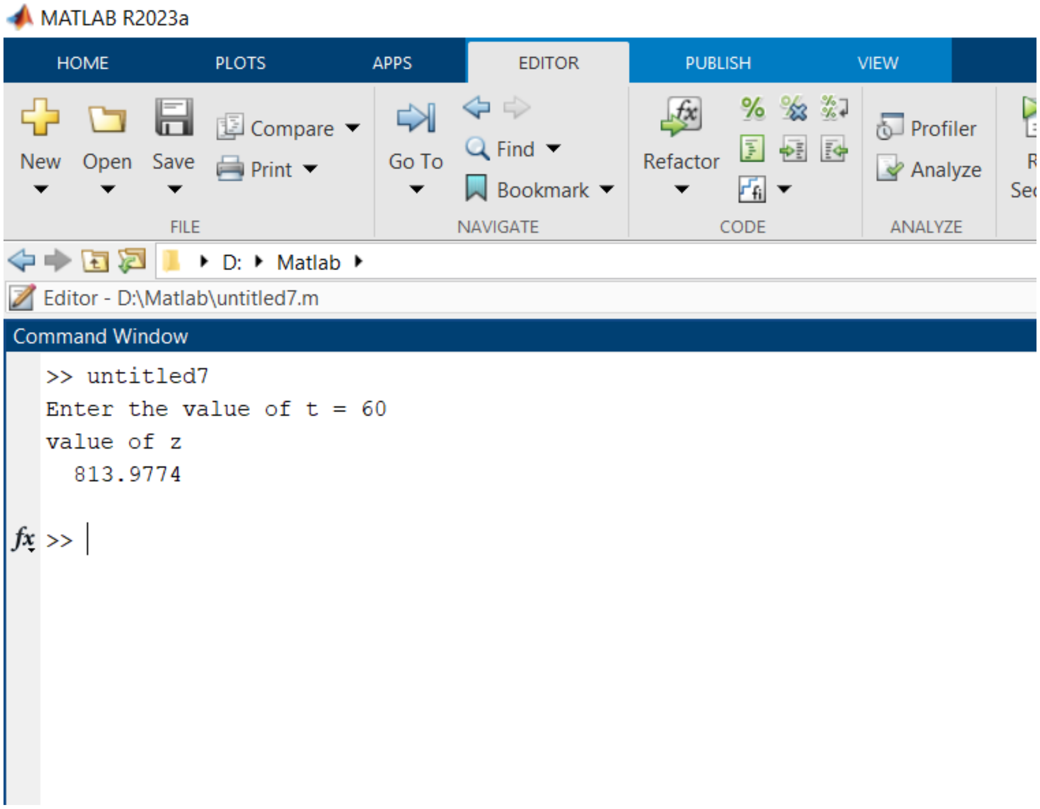
```
>> untitled6
Enter the value of u = 4
Enter the value of v = 6
value of y
    8.9287

fx >>
```

(3) $z = \sqrt{5e^{-0.2t}\cos 2t}$

SOURCE CODE:

```
% Define the values of t
t = input('Enter the value of t = ');
x = exp(0.2*t);

% Calculate the expression
z = sqrt(5*x*cos(2*t));

% Display the result
disp('value of z ');
disp(z)
```

OUTPUT:



```
>> untitled7
Enter the value of t = 60
value of z
   813.9774

fx >> 
```

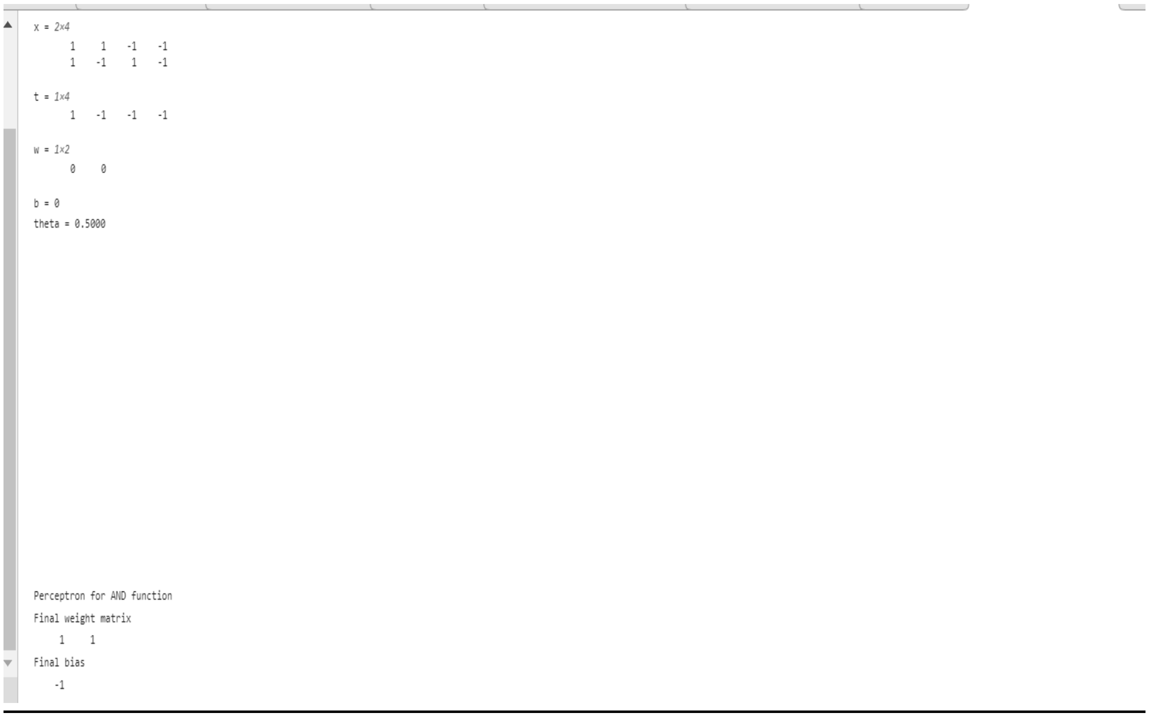
PROGRAM- 23

Aim - Write a MATLAB program for perceptron net for an AND function with bipolar inputs and targets.

SOURCE CODE:

```
%Perceptron for AND function
clear;
clc
x=[1 1 -1 -1;1 -1 1 -1]
t=[1 -1 -1 -1]
w=[0 0]
b=0
alpha=input('Enter Learning rate=');
theta=input('Enter Threshold value=')
con=1;
epoch=0;
while con
    con=0;
    for i=1:4
        yin=b+x(1,i)*w(1)+x(2,i)*w(2);
        if yin>theta
            y=1;
        end
        if yin<=theta & yin>=-theta
            y=0;
        end
        if yin<-theta
            y=-1;
        end
        if y-t(i)
            con=1;
            for j=1:2
                w(j)=w(j)+alpha*t(i)*x(j,i);
            end
            b=b+alpha*t(i);
        end
    end
end
disp('Perceptron for AND function');
disp('Final weight matrix');
disp(w);
disp('Final bias');
disp(b);
```

OUTPUT:



```
x = 2x4

       1     1    -1    -1
       1    -1     1    -1

t = 1x4

       1    -1    -1    -1

w = 1x2

       0     0

b = 0
theta = 0.5000

Perceptron for AND function
Final weight matrix
     1     1
Final bias
    -1
```