NELSON MANDELA AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY

(NM-AIST)

SCHOOL OF COMPUTATIONAL AND COMMUNICATION SCIENCE AND ENGINEERING (COCSE)

MATLAB PRACTICES

LECTURER: DR. MUSSA DIDA

COURSE NAME: MOBILE TELECOMMUNICATION TECHNOLOGY

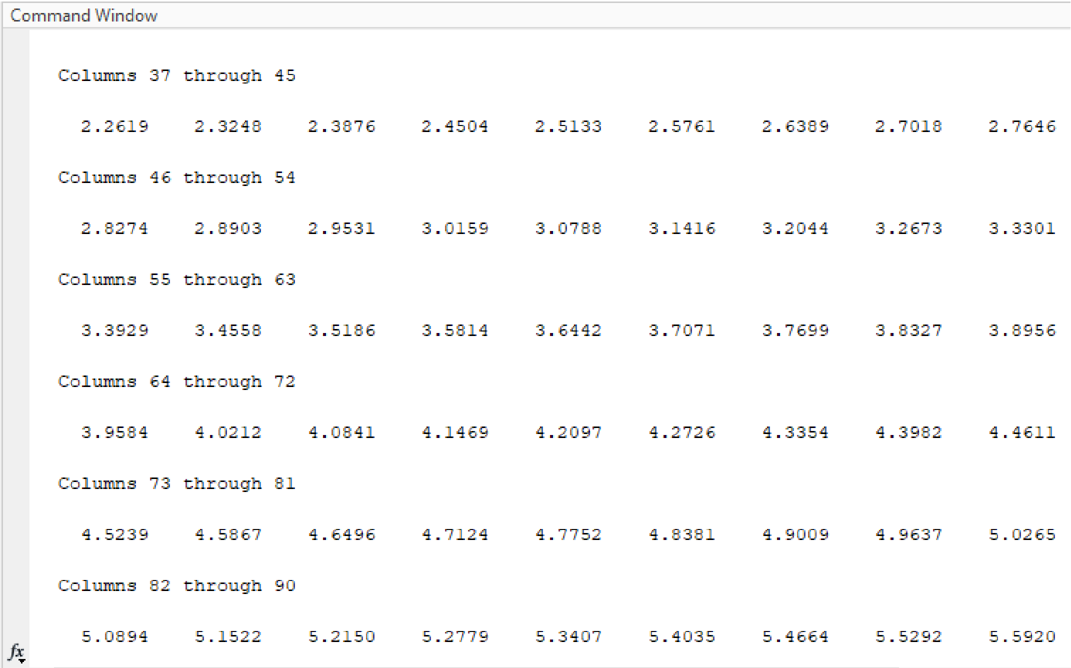
ASSIGNMENT: GROUP ASSIGNMENT TWO

COURSE CODE: EMoS 6222

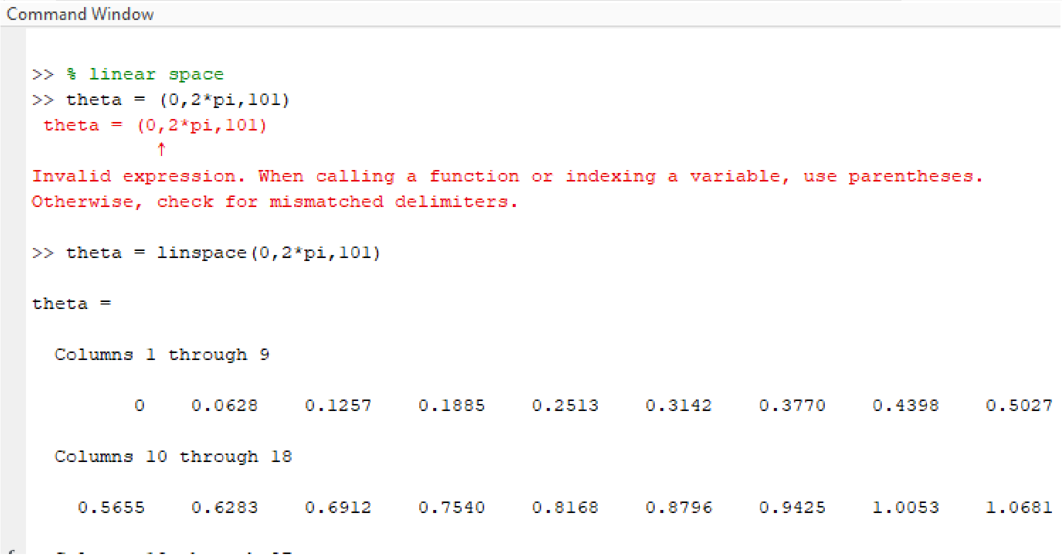
GROUP MEMBERS:

S/N	STUDENT NAME	REG. NUMBER	EMAIL
1.	CARMEL NKESHIMANA	M068/BI21	nkeshimanac@nm-aist.ac.tz
2	ELINAIKE O. MATHERU	M004/T21	matherue@nm-aist.ac.tz

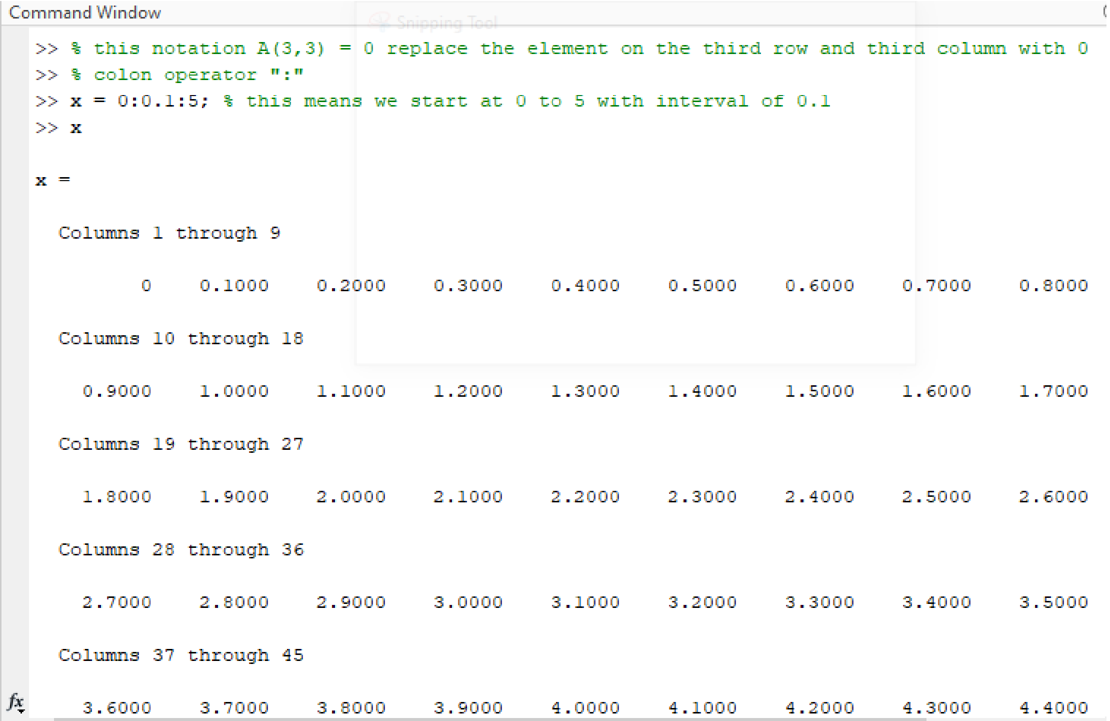
Examples: Matrix Arithmetic Operation



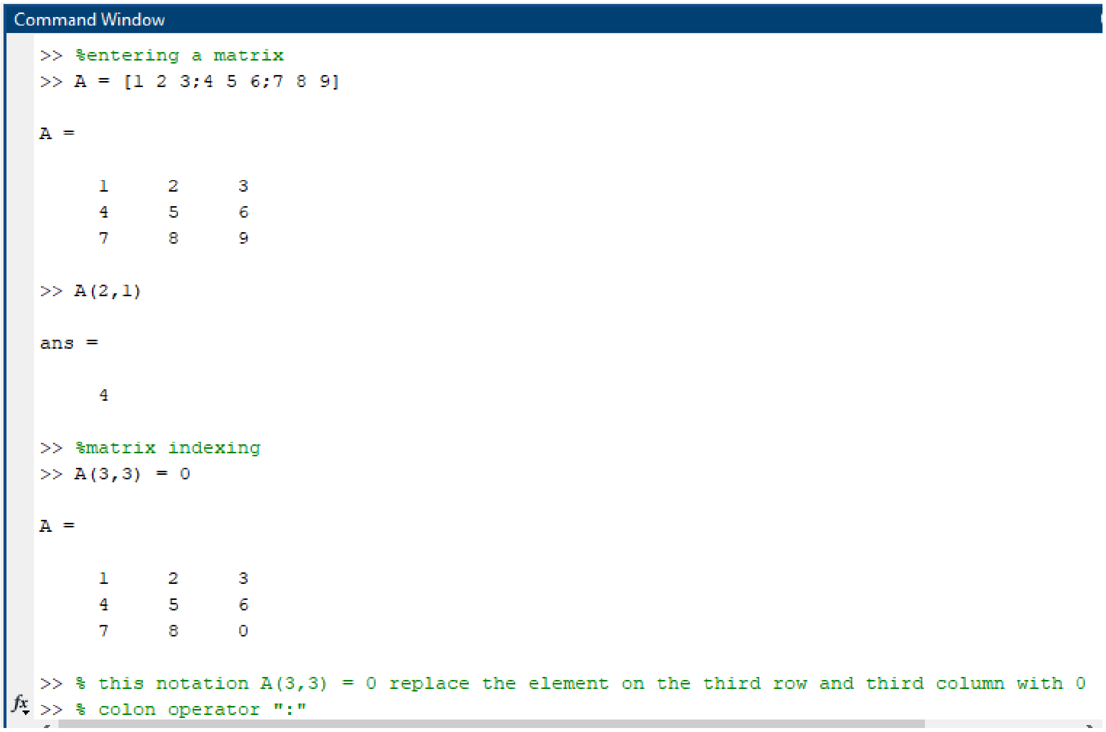
```
  Columns 37 through 45

    2.2619    2.3248    2.3876    2.4504    2.5133    2.5761    2.6389    2.7018    2.7646

  Columns 46 through 54

    2.8274    2.8903    2.9531    3.0159    3.0788    3.1416    3.2044    3.2673    3.3301

  Columns 55 through 63

    3.3929    3.4558    3.5186    3.5814    3.6442    3.7071    3.7699    3.8327    3.8956

  Columns 64 through 72

    3.9584    4.0212    4.0841    4.1469    4.2097    4.2726    4.3354    4.3982    4.4611

  Columns 73 through 81

    4.5239    4.5867    4.6496    4.7124    4.7752    4.8381    4.9009    4.9637    5.0265

  Columns 82 through 90

    5.0894    5.1522    5.2150    5.2779    5.3407    5.4035    5.4664    5.5292    5.5920
```



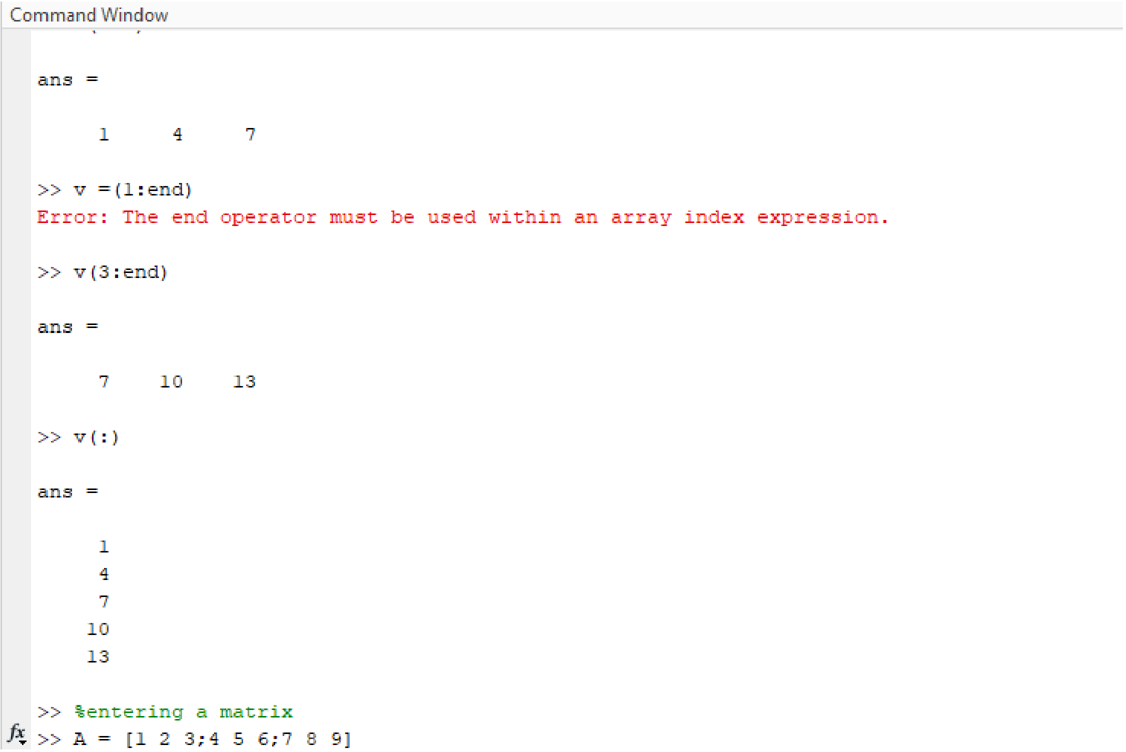
```
>> % linear space
>> theta = (0,2*pi,101)
 theta = (0,2*pi,101)
          ↑
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

>> theta = linspace(0,2*pi,101)

theta =

  Columns 1 through 9

         0    0.0628    0.1257    0.1885    0.2513    0.3142    0.3770    0.4398    0.5027

  Columns 10 through 18

    0.5655    0.6283    0.6912    0.7540    0.8168    0.8796    0.9425    1.0053    1.0681
```



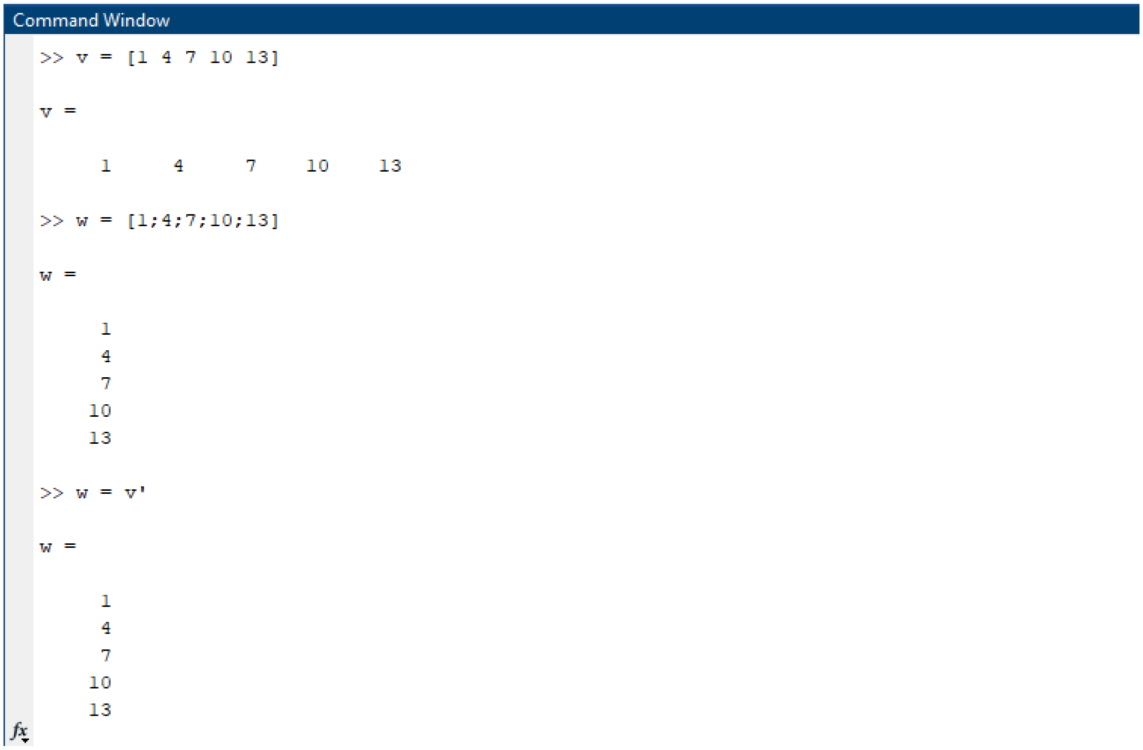
```
>> % this notation A(3,3) = 0 replace the element on the third row and third column with 0
>> % colon operator ":"
>> x = 0:0.1:5; % this means we start at 0 to 5 with interval of 0.1
>> x

x =

  Columns 1 through 9

         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000    0.8000

  Columns 10 through 18

    0.9000    1.0000    1.1000    1.2000    1.3000    1.4000    1.5000    1.6000    1.7000

  Columns 19 through 27

    1.8000    1.9000    2.0000    2.1000    2.2000    2.3000    2.4000    2.5000    2.6000

  Columns 28 through 36

    2.7000    2.8000    2.9000    3.0000    3.1000    3.2000    3.3000    3.4000    3.5000

  Columns 37 through 45

    3.6000    3.7000    3.8000    3.9000    4.0000    4.1000    4.2000    4.3000    4.4000
```



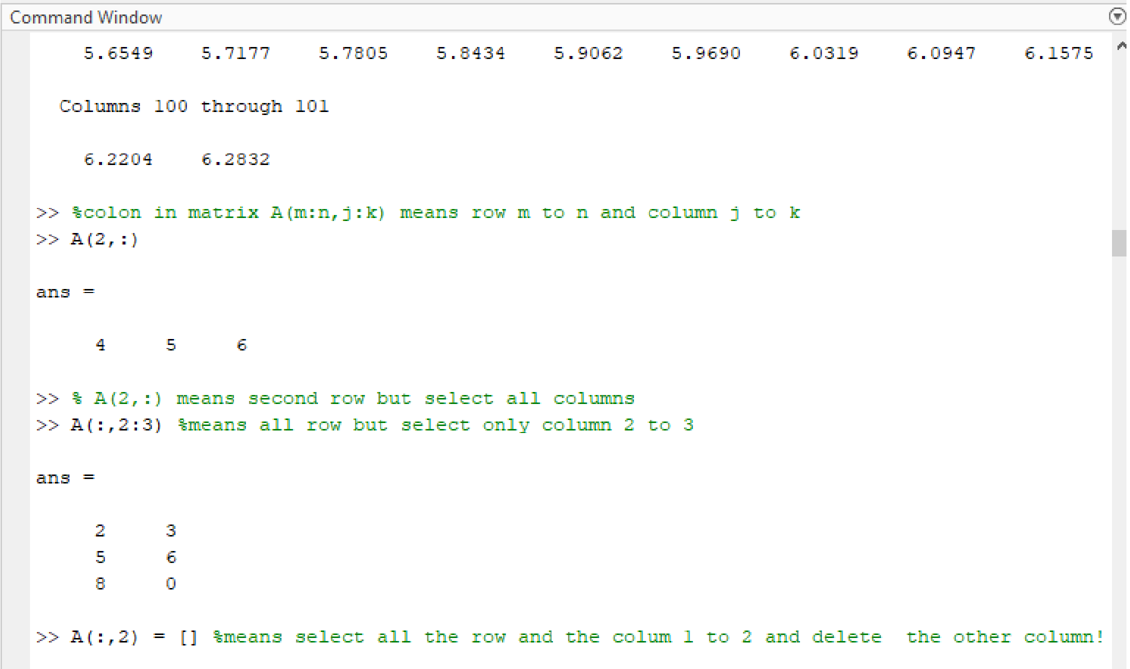
```
>> %entering a matrix
>> A = [1 2 3;4 5 6;7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9

>> A(2,1)

ans =

     4

>> %matrix indexing
>> A(3,3) = 0

A =

     1     2     3
     4     5     6
     7     8     0

>> % this notation A(3,3) = 0 replace the element on the third row and third column with 0
>> % colon operator ":"
```



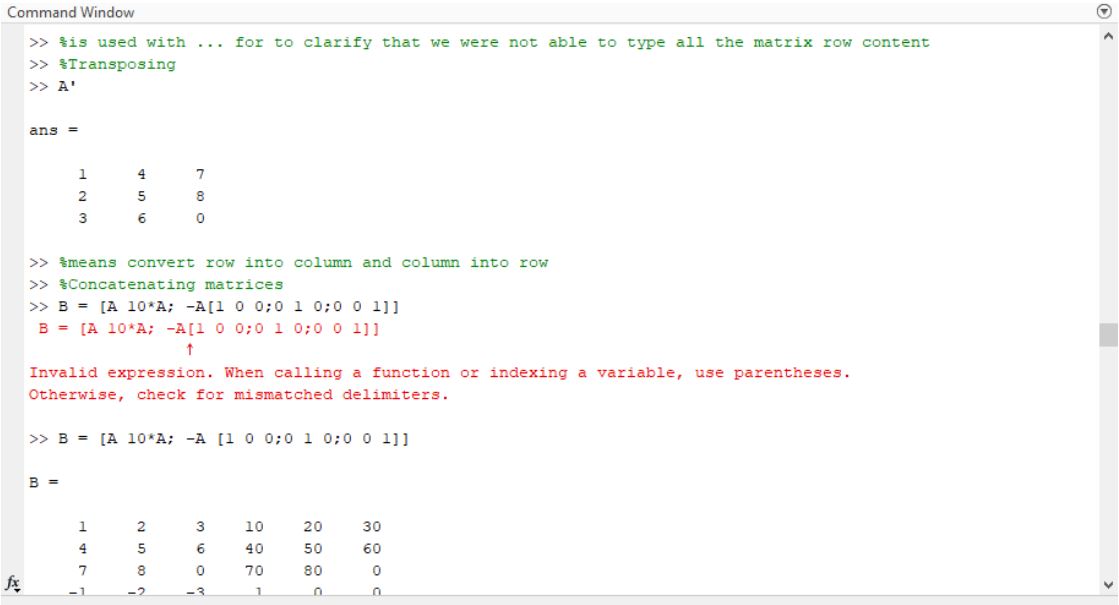
```
ans =

     1     4     7

>> v =(1:end)
Error: The end operator must be used within an array index expression.

>> v(3:end)

ans =

     7    10    13

>> v(:)

ans =

     1
     4
     7
    10
    13

>> %entering a matrix
>> A = [1 2 3;4 5 6;7 8 9]
```



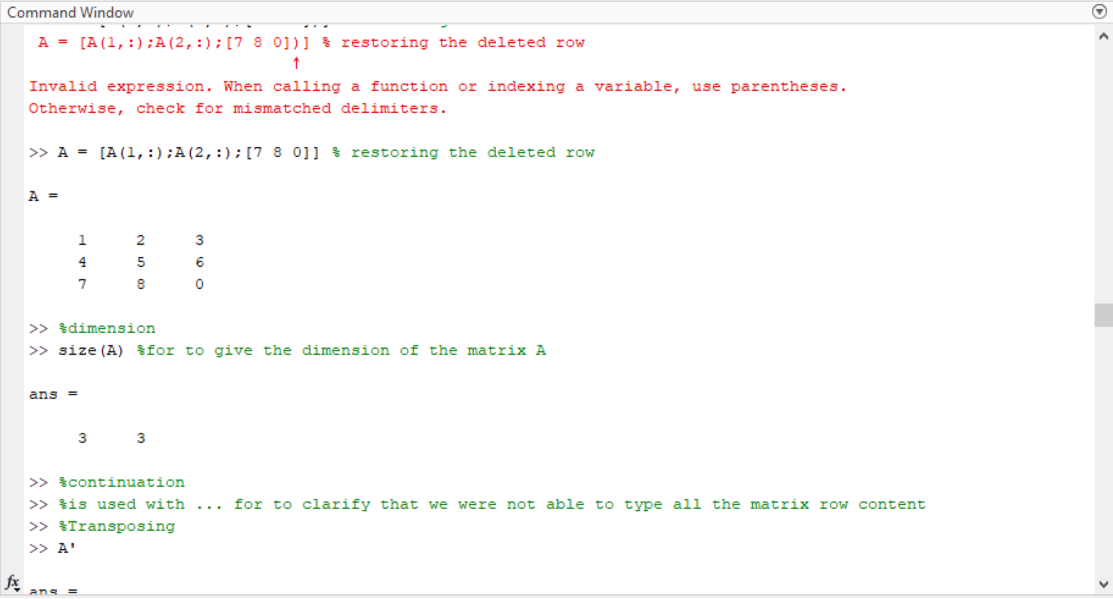
```
>> v = [1 4 7 10 13]

v =

     1     4     7    10    13

>> w = [1;4;7;10;13]

w =

     1
     4
     7
    10
    13

>> w = v'

w =

     1
     4
     7
    10
    13
```



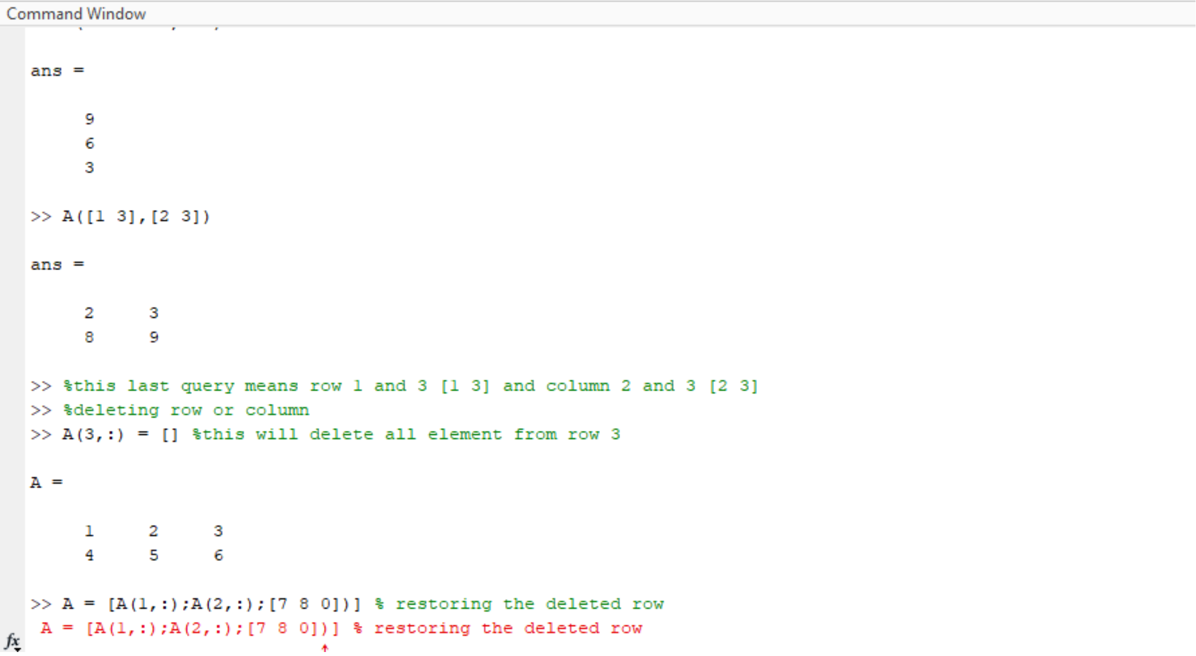
```
    5.6549    5.7177    5.7805    5.8434    5.9062    5.9690    6.0319    6.0947    6.1575

  Columns 100 through 101

    6.2204    6.2832

>> %colon in matrix A(m:n,j:k) means row m to n and column j to k
>> A(2,:)

ans =

     4     5     6

>> % A(2,:) means second row but select all columns
>> A(:,2:3) %means all row but select only column 2 to 3

ans =

     2     3
     5     6
     8     0

>> A(:,2) = [] %means select all the row and the colum 1 to 2 and delete  the other column!
```



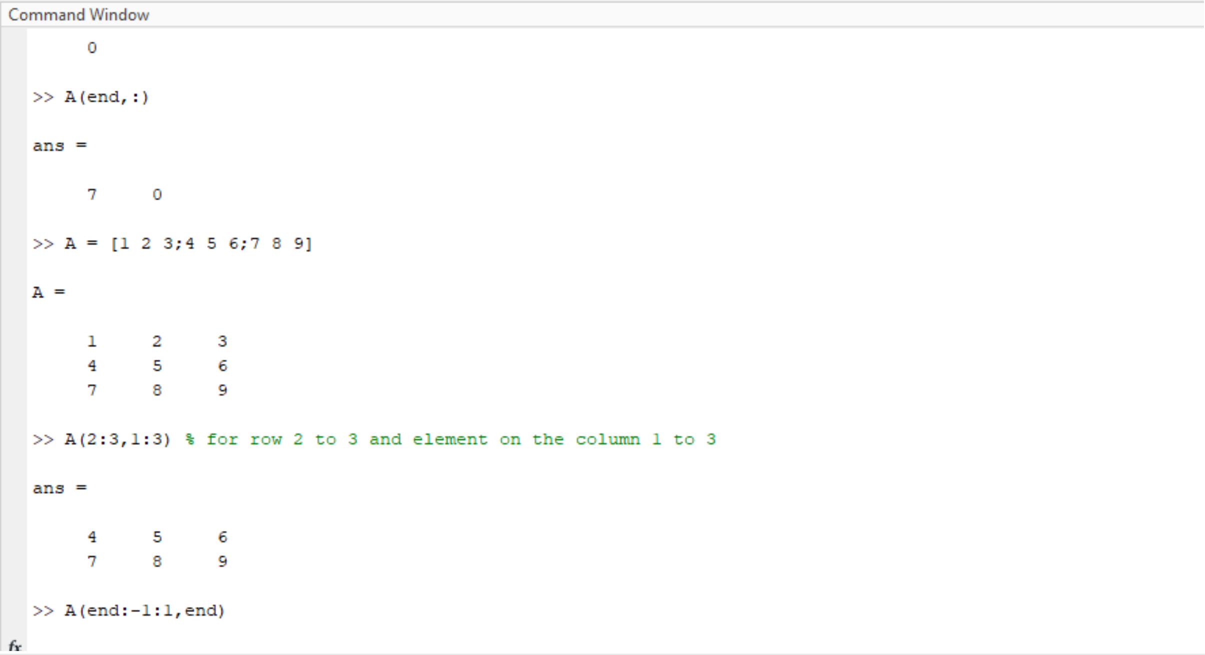
```
>> %is used with ... for to clarify that we were not able to type all the matrix row content
>> %Transposing
>> A'

ans =

     1     4     7
     2     5     8
     3     6     0

>> %means convert row into column and column into row
>> %Concatenating matrices
>> B = [A 10*A; -A[1 0 0;0 1 0;0 0 1]]
 B = [A 10*A; -A[1 0 0;0 1 0;0 0 1]]
                ↑
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

>> B = [A 10*A; -A [1 0 0;0 1 0;0 0 1]]

B =

     1     2     3    10    20    30
     4     5     6    40    50    60
     7     8     0    70    80     0
    -1    -2    -3     1     0     0
```



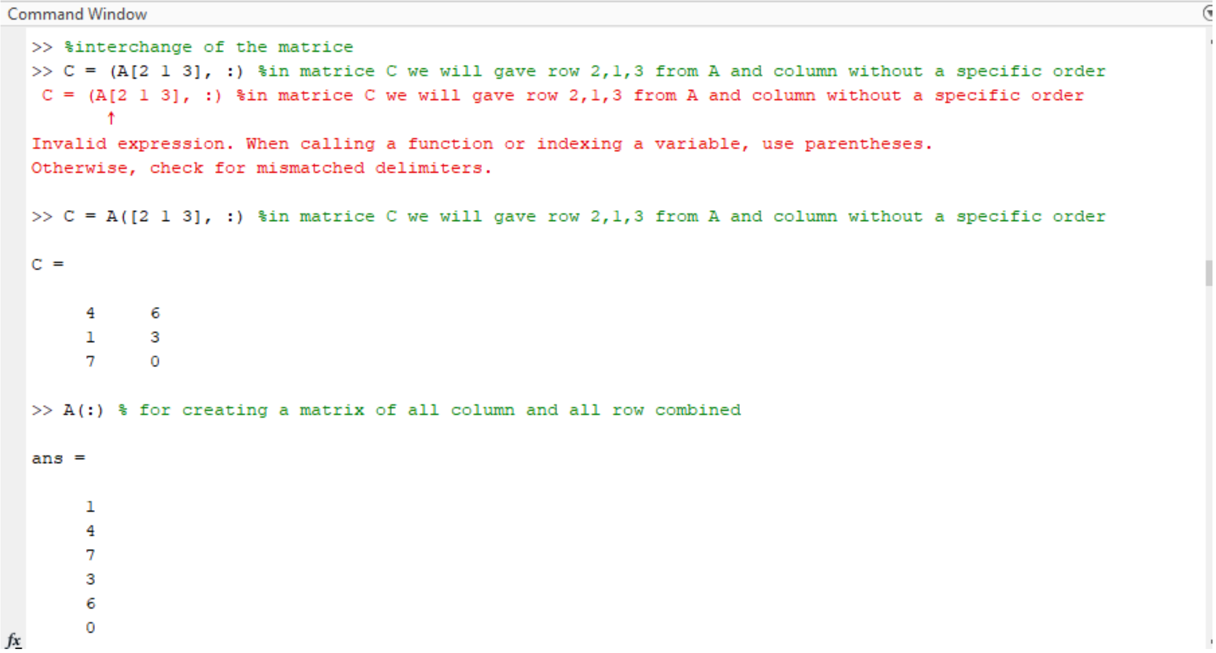
```
 A = [A(1,:);A(2,:);[7 8 0])] % restoring the deleted row
                               ↑
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

>> A = [A(1,:);A(2,:);[7 8 0]] % restoring the deleted row

A =

     1     2     3
     4     5     6
     7     8     0

>> %dimension
>> size(A) %for to give the dimension of the matrix A

ans =

     3     3

>> %continuation
>> %is used with ... for to clarify that we were not able to type all the matrix row content
>> %Transposing
>> A'

ans =
```



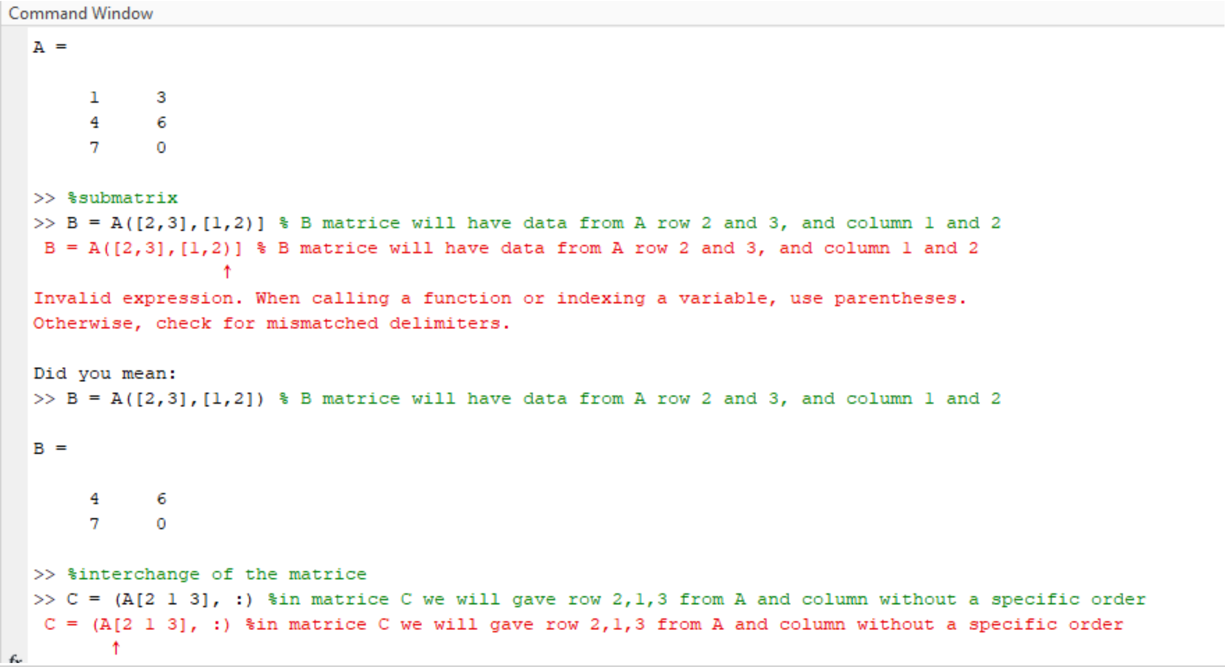
```
ans =

     9
     6
     3

>> A([1 3],[2 3])

ans =

     2     3
     8     9

>> %this last query means row 1 and 3 [1 3] and column 2 and 3 [2 3]
>> %deleting row or column
>> A(3,:) = [] %this will delete all element from row 3

A =

     1     2     3
     4     5     6

>> A = [A(1,:);A(2,:);[7 8 0])] % restoring the deleted row
 A = [A(1,:);A(2,:);[7 8 0])] % restoring the deleted row
                               ↑
```



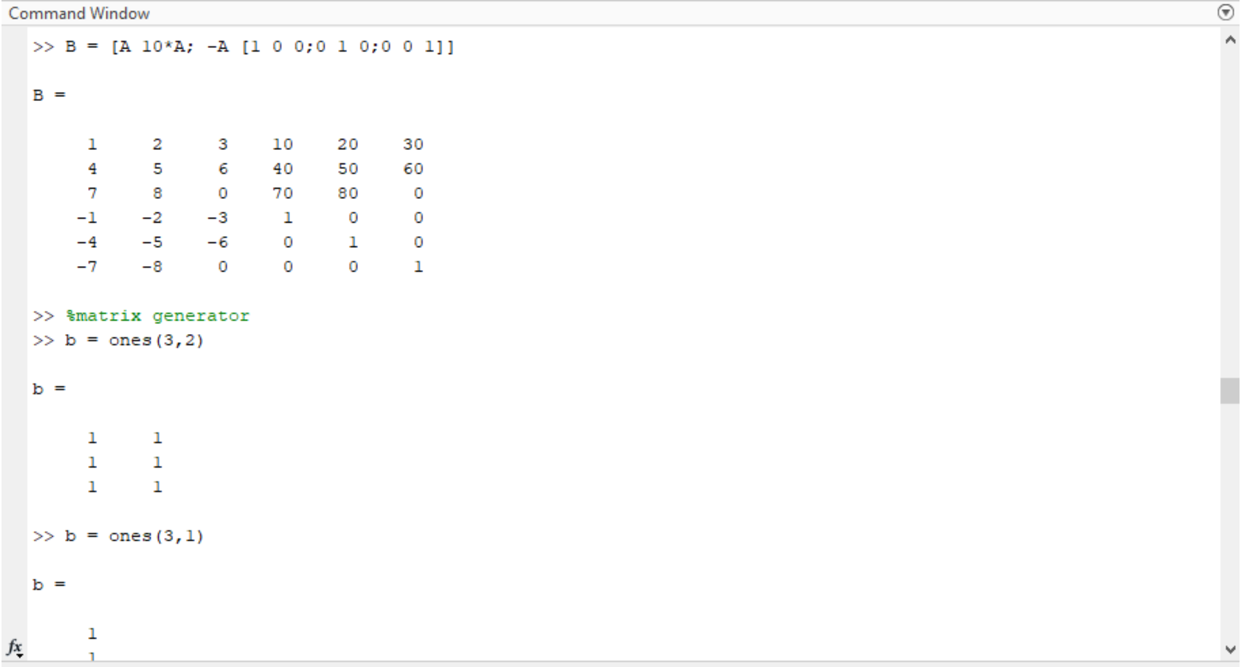
```
     0

>> A(end,:)

ans =

     7     0

>> A = [1 2 3;4 5 6;7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9

>> A(2:3,1:3) % for row 2 to 3 and element on the column 1 to 3

ans =

     4     5     6
     7     8     9

>> A(end:-1:1,end)
```



```
>> %interchange of the matrice
>> C = (A[2 1 3], :) %in matrice C we will gave row 2,1,3 from A and column without a specific order
 C = (A[2 1 3], :) %in matrice C we will gave row 2,1,3 from A and column without a specific order
       ↑
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

>> C = A([2 1 3], :) %in matrice C we will gave row 2,1,3 from A and column without a specific order

C =

     4     6
     1     3
     7     0

>> A(:) % for creating a matrix of all column and all row combined

ans =

     1
     4
     7
     3
     6
     0
```

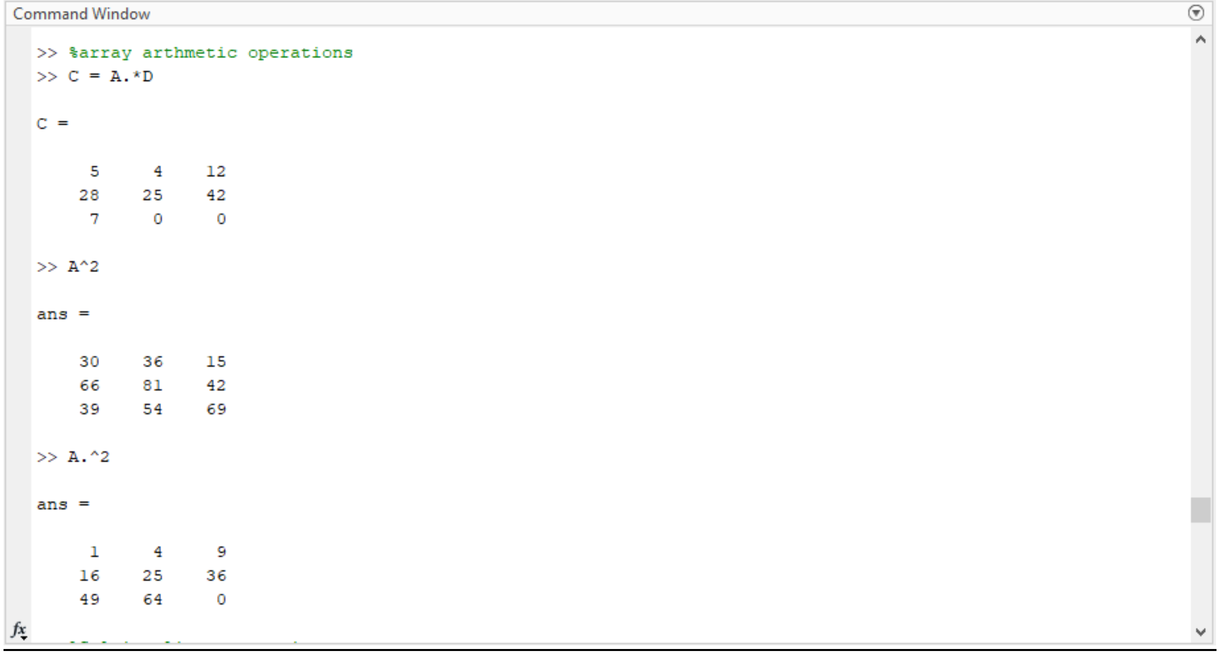


```
A =

     1     3
     4     6
     7     0

>> %submatrix
>> B = A([2,3],[1,2)] % B matrice will have data from A row 2 and 3, and column 1 and 2
 B = A([2,3],[1,2)] % B matrice will have data from A row 2 and 3, and column 1 and 2
                 ↑
Invalid expression. When calling a function or indexing a variable, use parentheses.
Otherwise, check for mismatched delimiters.

Did you mean:
>> B = A([2,3],[1,2]) % B matrice will have data from A row 2 and 3, and column 1 and 2

B =

     4     6
     7     0

>> %interchange of the matrice
>> C = (A[2 1 3], :) %in matrice C we will gave row 2,1,3 from A and column without a specific order
 C = (A[2 1 3], :) %in matrice C we will gave row 2,1,3 from A and column without a specific order
       ↑
```

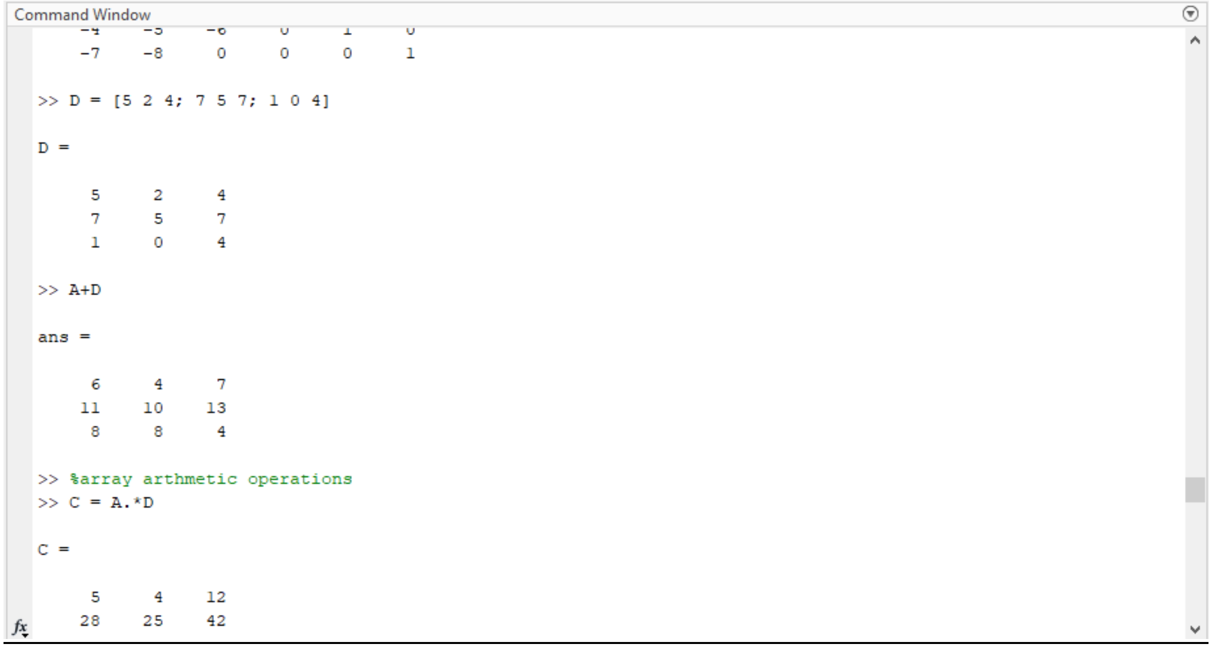


```
>> B = [A 10*A; -A [1 0 0;0 1 0;0 0 1]]

B =

     1     2     3    10    20    30
     4     5     6    40    50    60
     7     8     0    70    80     0
    -1    -2    -3     1     0     0
    -4    -5    -6     0     1     0
    -7    -8     0     0     0     1

>> %matrix generator
>> b = ones(3,2)

b =

     1     1
     1     1
     1     1

>> b = ones(3,1)

b =

     1
     1
```

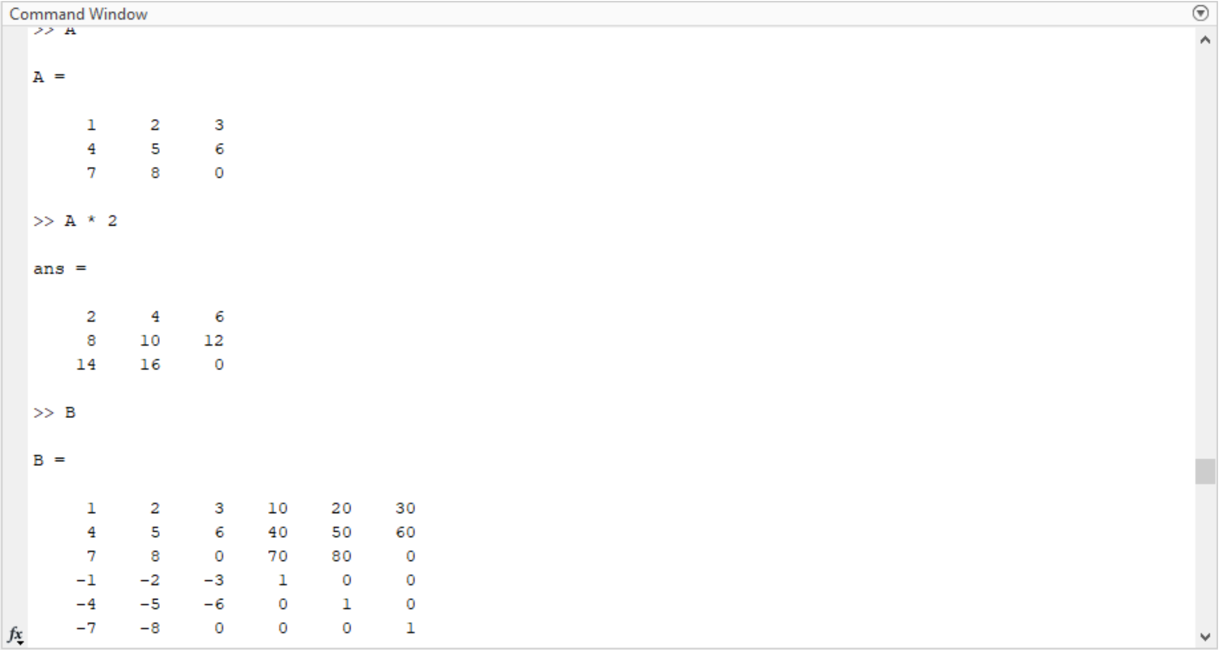
Examples: Array Operation & Linear Equations



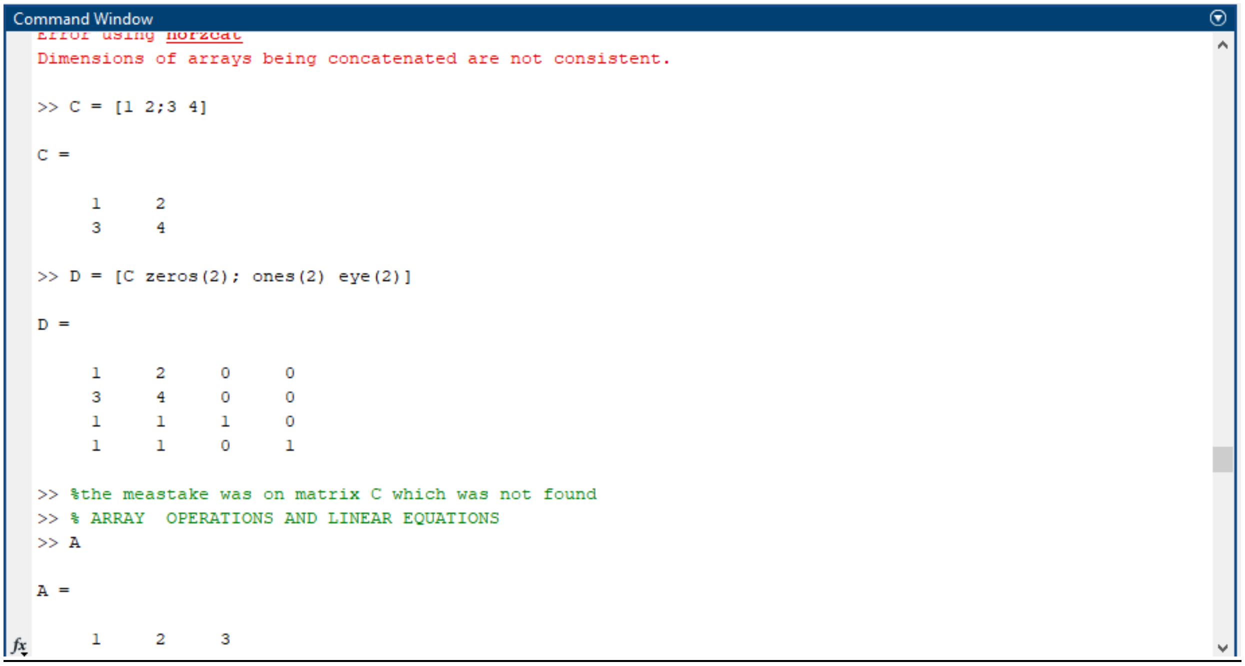
```
>> %array arthmetic operations
>> C = A.*D

C =

     5     4    12
    28    25    42
     7     0     0

>> A^2

ans =

    30    36    15
    66    81    42
    39    54    69

>> A.^2

ans =

     1     4     9
    16    25    36
    49    64     0
```



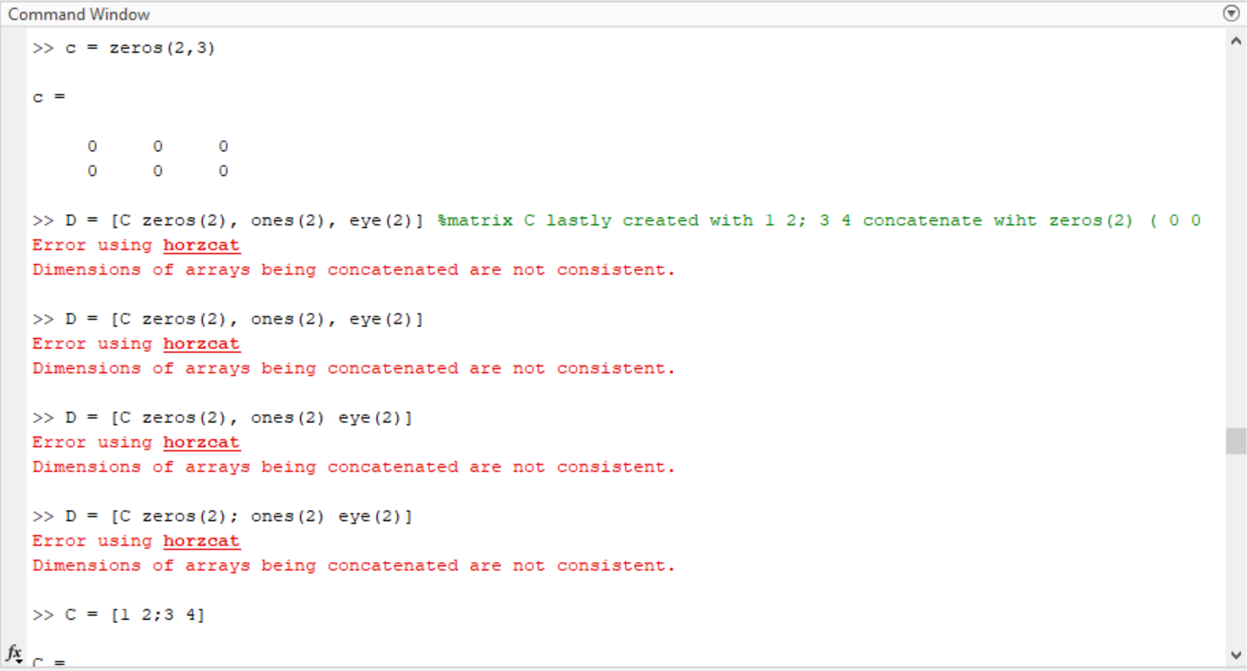
```
    -4    -5    -6     0     1     0
    -7    -8     0     0     0     1

>> D = [5 2 4; 7 5 7; 1 0 4]

D =

     5     2     4
     7     5     7
     1     0     4

>> A+D

ans =

     6     4     7
    11    10    13
     8     8     4

>> %array arthmetic operations
>> C = A.*D

C =

     5     4    12
    28    25    42
```



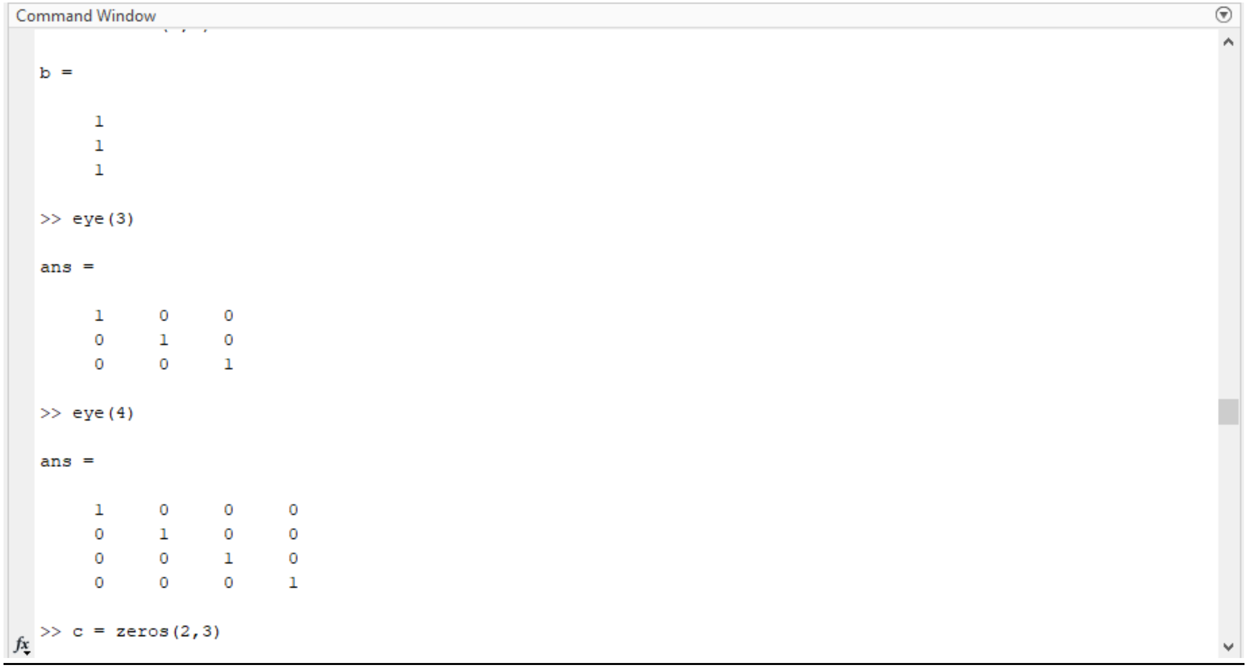
```
>> A

A =

     1     2     3
     4     5     6
     7     8     0

>> A * 2

ans =

     2     4     6
     8    10    12
    14    16     0

>> B

B =

     1     2     3    10    20    30
     4     5     6    40    50    60
     7     8     0    70    80     0
    -1    -2    -3     1     0     0
    -4    -5    -6     0     1     0
    -7    -8     0     0     0     1
```



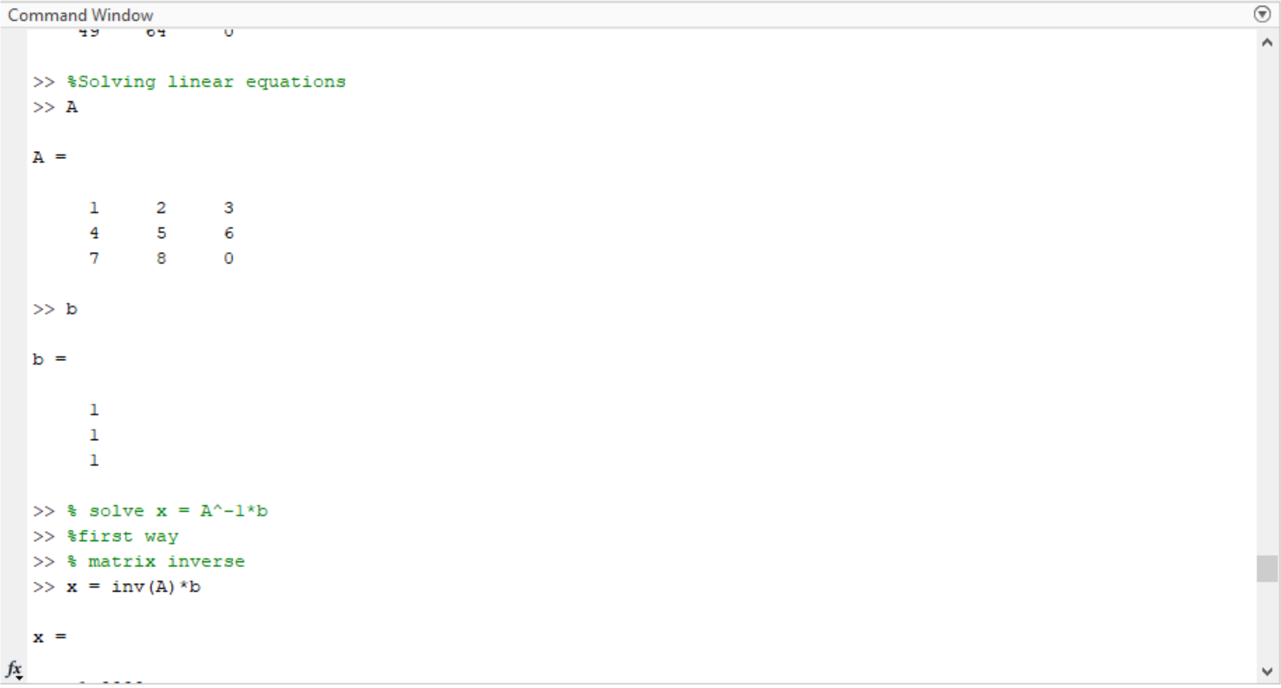
```
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> C = [1 2;3 4]

C =

     1     2
     3     4

>> D = [C zeros(2); ones(2) eye(2)]

D =

     1     2     0     0
     3     4     0     0
     1     1     1     0
     1     1     0     1

>> %the meastake was on matrix C which was not found
>> % ARRAY  OPERATIONS AND LINEAR EQUATIONS
>> A

A =

     1     2     3
```



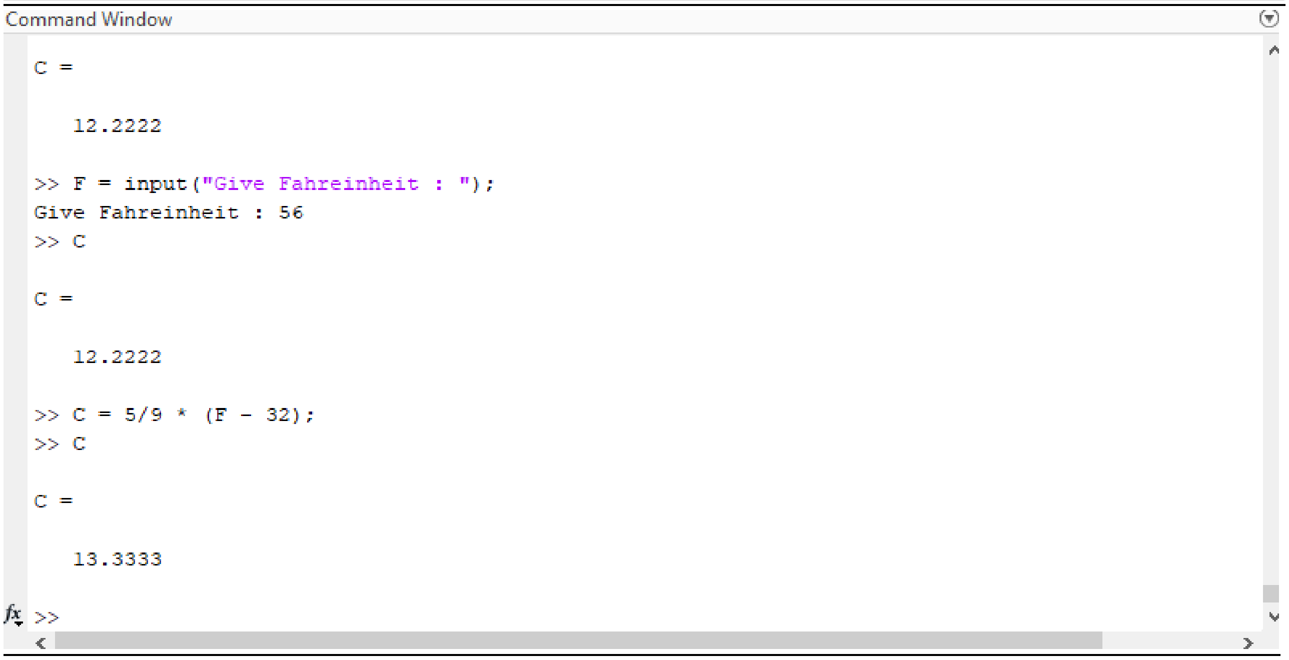
```
>> c = zeros(2,3)

c =

     0     0     0
     0     0     0

>> D = [C zeros(2), ones(2), eye(2)] %matrix C lastly created with 1 2; 3 4 concatenate wiht zeros(2)  ( 0 0
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> D = [C zeros(2), ones(2), eye(2)]
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> D = [C zeros(2), ones(2) eye(2)]
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> D = [C zeros(2); ones(2) eye(2)]
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> C = [1 2;3 4]

C =
```



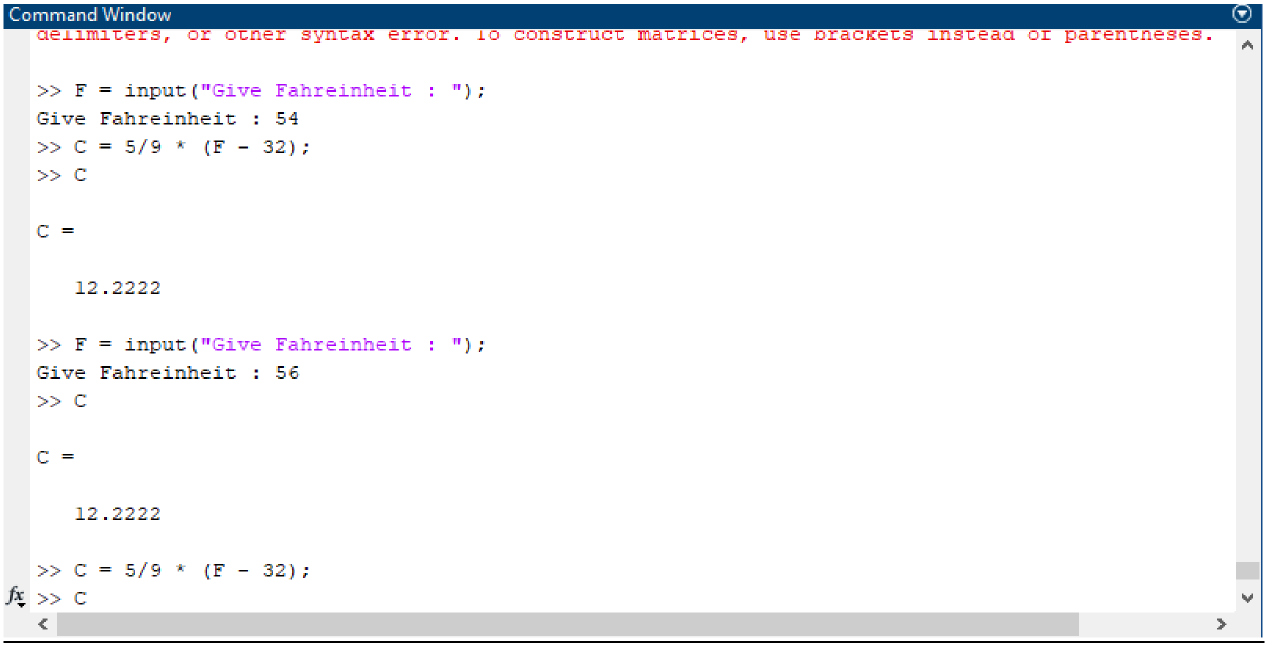
```
b =

     1
     1
     1

>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1

>> eye(4)

ans =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> c = zeros(2,3)
```



```
    49    64     0

>> %Solving linear equations
>> A

A =

     1     2     3
     4     5     6
     7     8     0

>> b

b =

     1
     1
     1

>> % solve x = A^-1*b
>> %first way
>> % matrix inverse
>> x = inv(A)*b

x =
```

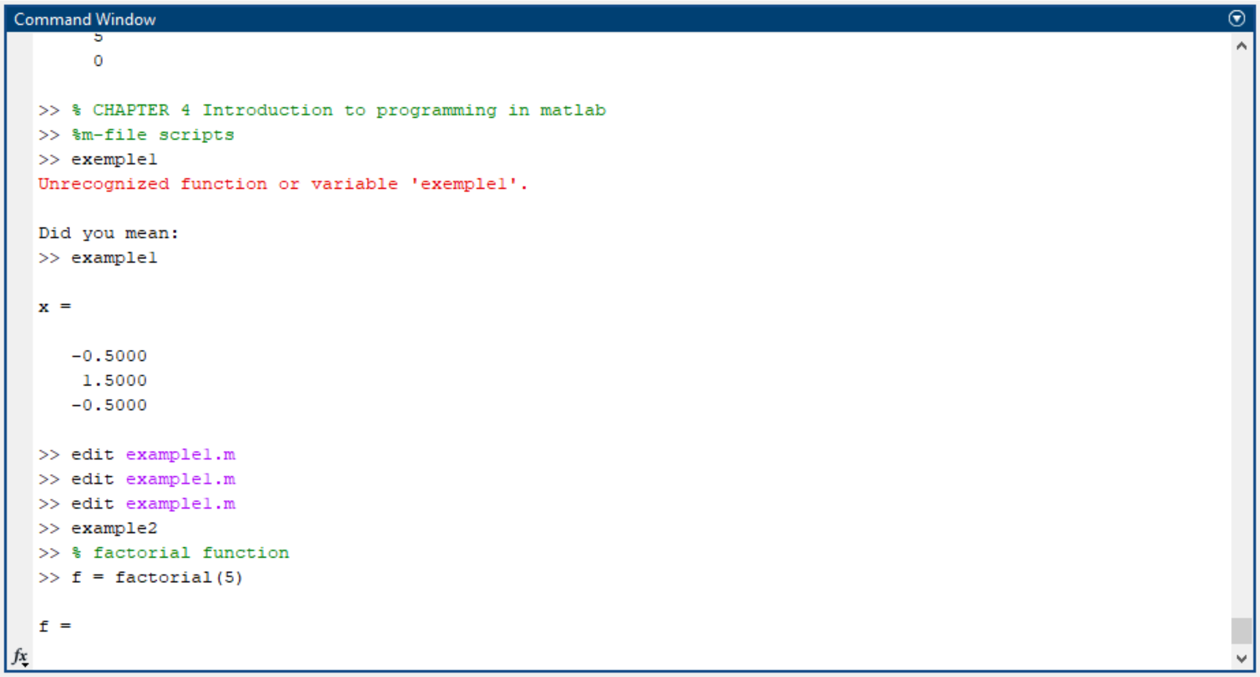


```
C =

   12.2222

>> F = input("Give Fahreinheit : ");
Give Fahreinheit : 56
>> C

C =

   12.2222

>> C = 5/9 * (F - 32);
>> C

C =

   13.3333

>>
```

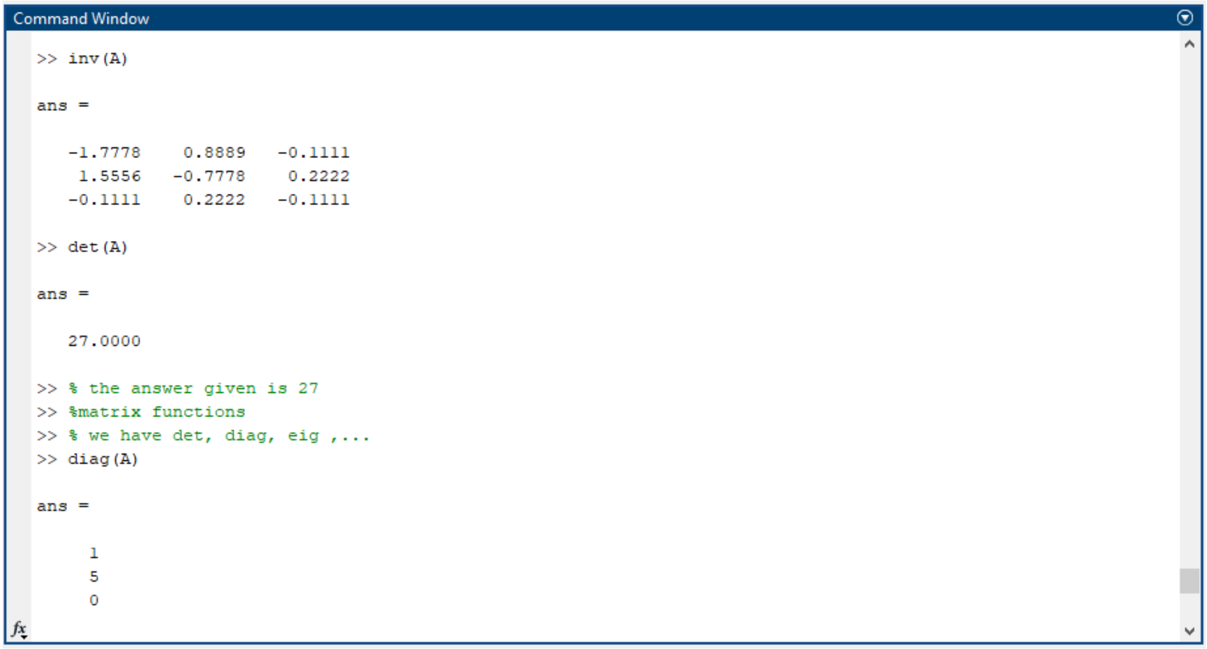


```
delimiters, or other syntax error. To construct matrices, use brackets instead of parentheses.

>> F = input("Give Fahreinheit : ");
Give Fahreinheit : 54
>> C = 5/9 * (F - 32);
>> C

C =

   12.2222

>> F = input("Give Fahreinheit : ");
Give Fahreinheit : 56
>> C

C =

   12.2222

>> C = 5/9 * (F - 32);
>> C
```

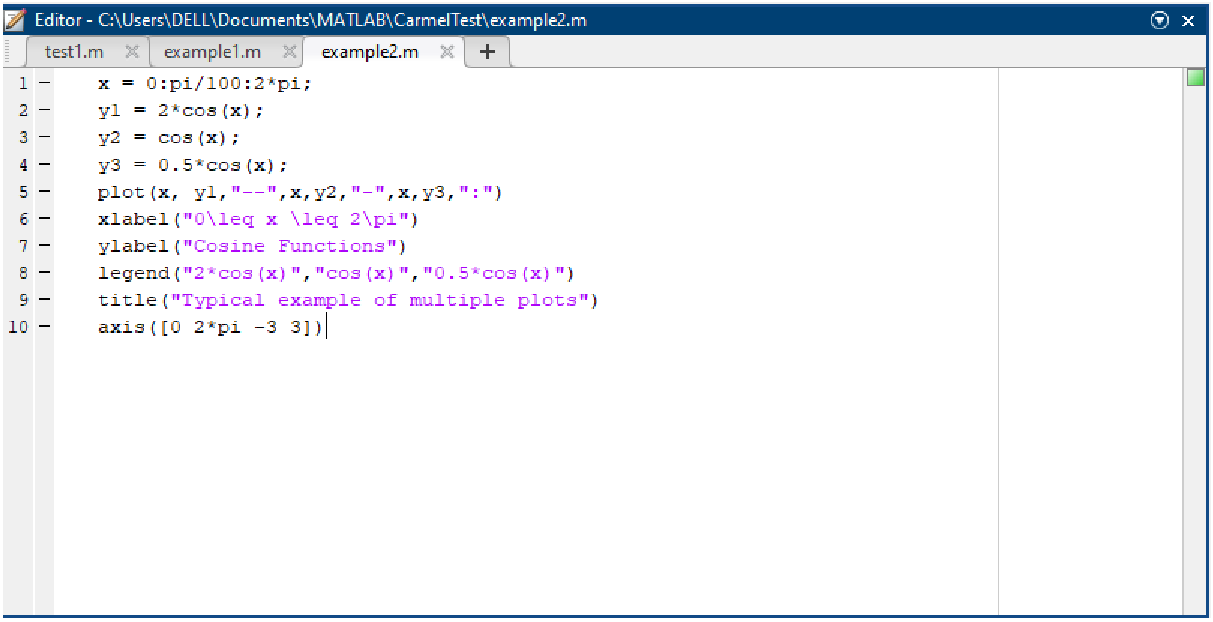
Examples: Introduction to Programming in MATLAB



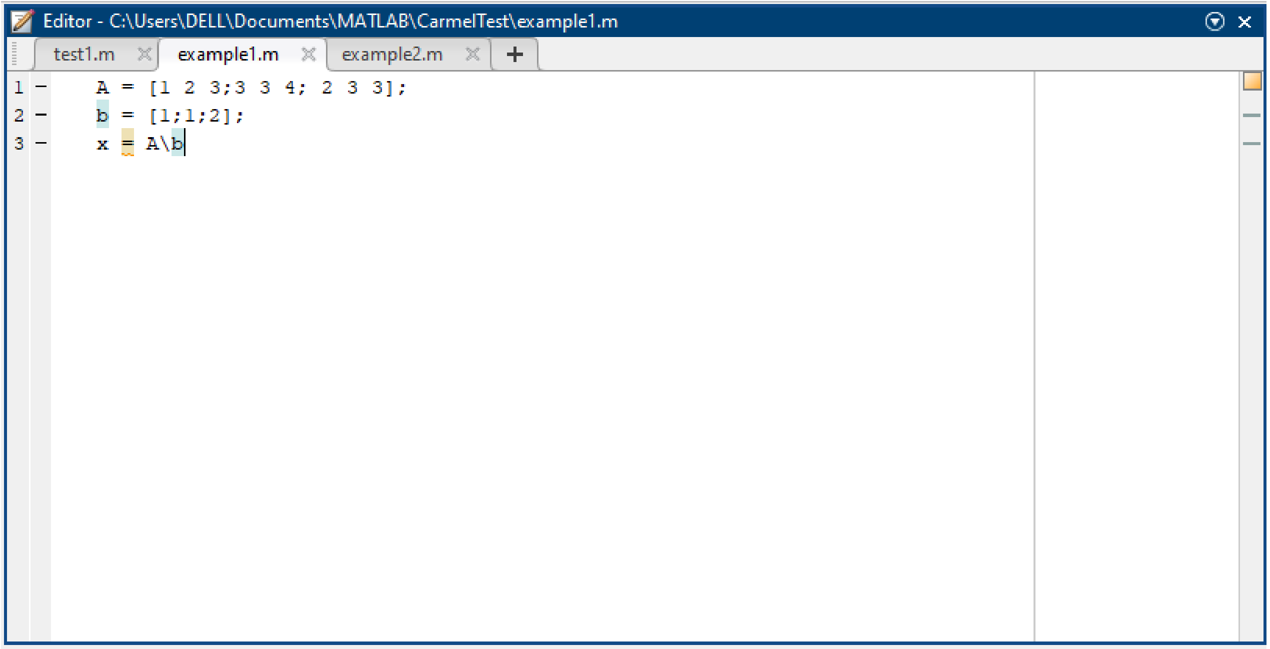
```
      5
      0

>> % CHAPTER 4 Introduction to programming in matlab
>> %m-file scripts
>> exemple1
Unrecognized function or variable 'exemple1'.

Did you mean:
>> example1

x =

   -0.5000
    1.5000
   -0.5000

>> edit example1.m
>> edit example1.m
>> edit example1.m
>> example2
>> % factorial function
>> f = factorial(5)

f =
```



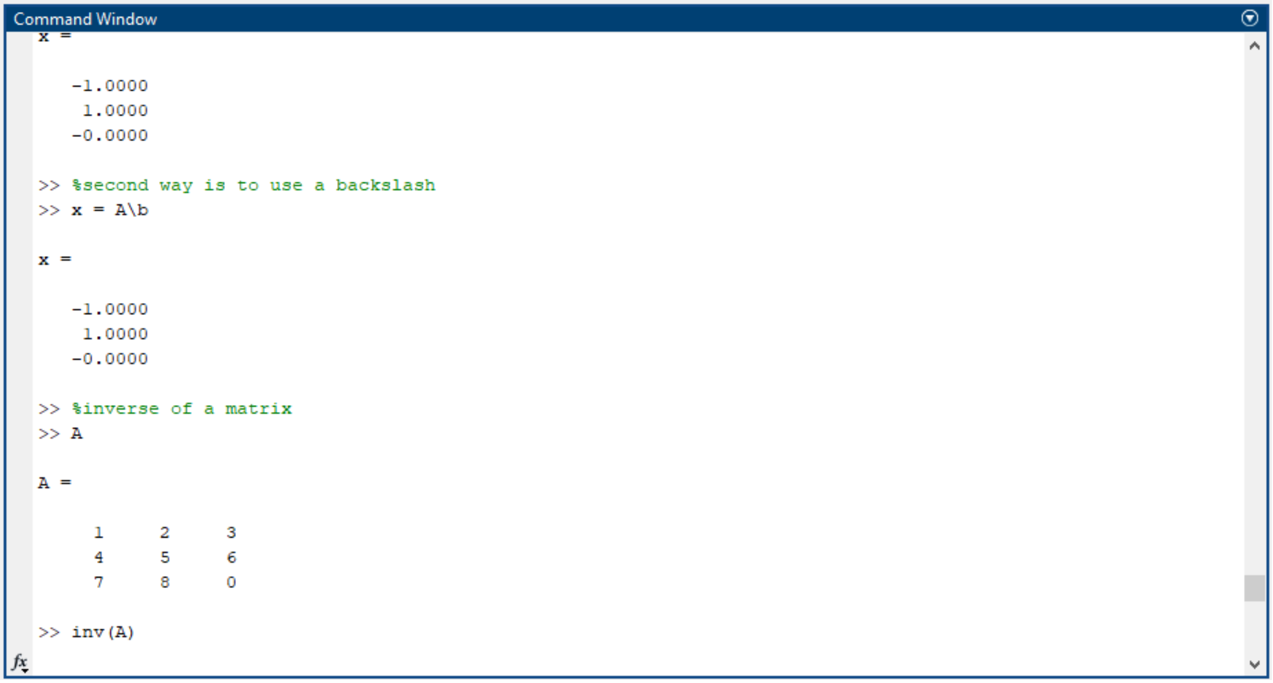
```
>> inv(A)

ans =

   -1.7778    0.8889   -0.1111
    1.5556   -0.7778    0.2222
   -0.1111    0.2222   -0.1111

>> det(A)

ans =

   27.0000

>> % the answer given is 27
>> %matrix functions
>> % we have det, diag, eig ,...
>> diag(A)

ans =

     1
     5
     0
```


Editor - C:\Users\DELL\Documents\MATLAB\CarmelTest\example2.m
test1.m
example1.m
example2.m
+

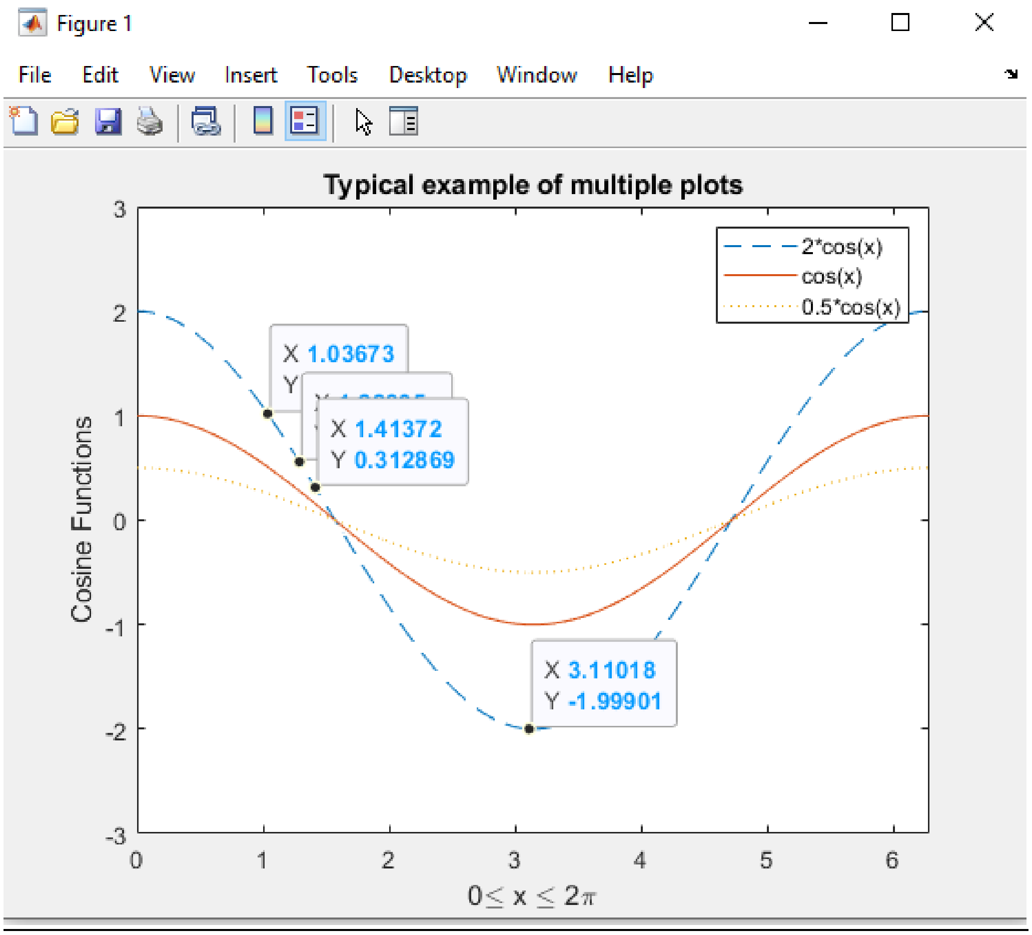
```
x = 0:pi/100:2*pi;
y1 = 2*cos(x);
y2 = cos(x);
y3 = 0.5*cos(x);
plot(x, y1,"--",x,y2,"-",x,y3,":")
xlabel("0\leq x \leq 2\pi")
ylabel("Cosine Functions")
legend("2*cos(x)","cos(x)","0.5*cos(x)")
title("Typical example of multiple plots")
axis([0 2*pi -3 3])
```


Editor - C:\Users\DELL\Documents\MATLAB\CarmelTest\example1.m
test1.m
example1.m
example2.m
+

```
A = [1 2 3;3 3 4; 2 3 3];
b = [1;1;2];
x = A\b
```

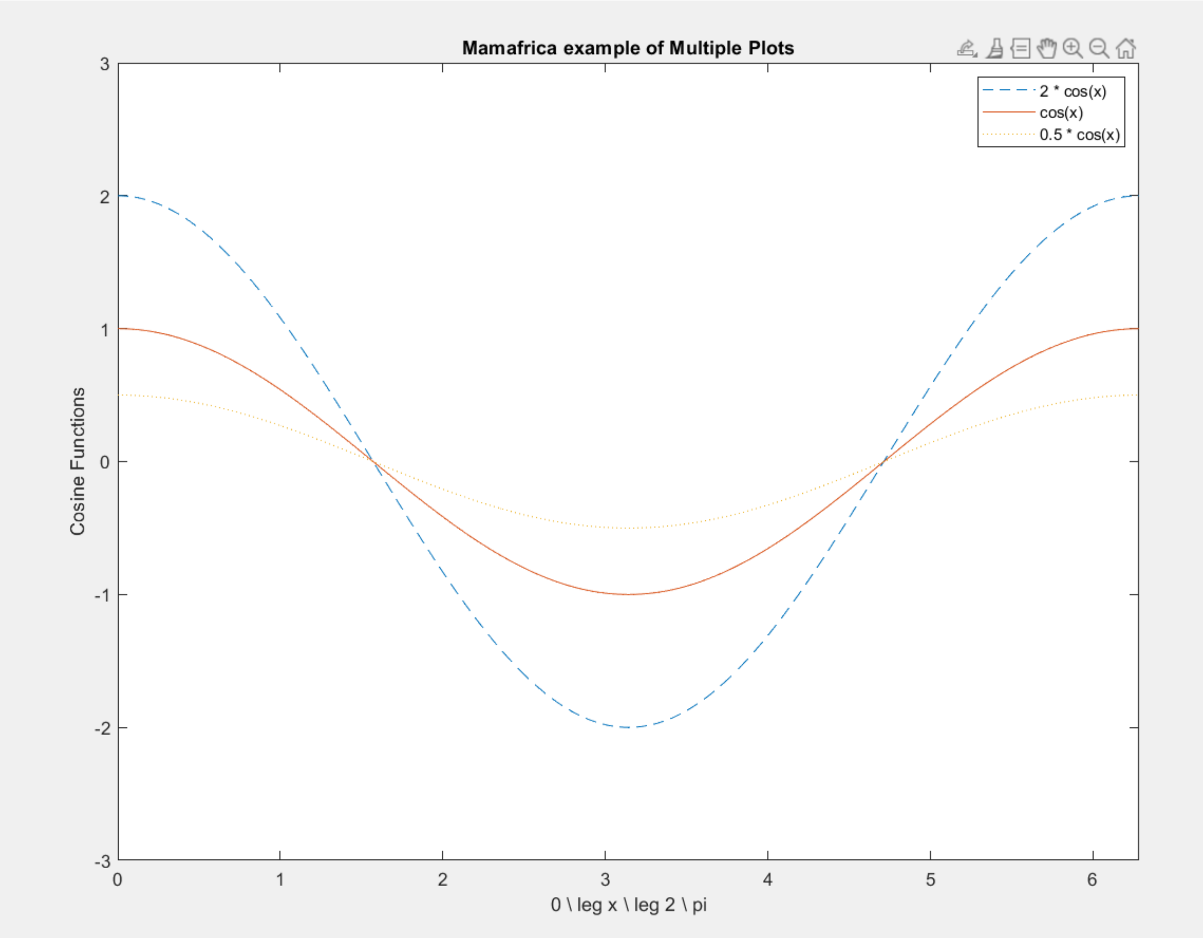


```
x =

   -1.0000
    1.0000
   -0.0000

>> %second way is to use a backslash
>> x = A\b

x =

   -1.0000
    1.0000
   -0.0000

>> %inverse of a matrix
>> A

A =

     1     2     3
     4     5     6
     7     8     0

>> inv(A)
```



Plot the following cosine functions, y1 = 2 cos(x), y2 = cos(x), and y3 = 0.5 ∗ cos(x), in the interval 0 ≤ x ≤ 2π. This example has been presented in previous Chapter. Here we put the commands in a file. • Create a file, say example2.m, which contains the following commands:

```
 x = 0:pi/100:2*pi;

y1 = 2*cos(x);

y2 = cos(x);

y3 = 0.5*cos(x);

plot(x,y1,’--’,x,y2,’-’,x,y3,’:’)

xlabel(’0 \leq x \leq 2\pi’)

ylabel(’Cosine functions’)

legend(’2*cos(x)’,’cos(x)’,’0.5*cos(x)’)

title(’Typical example of multiple plots’)

axis([0 2*pi -3 3])
```



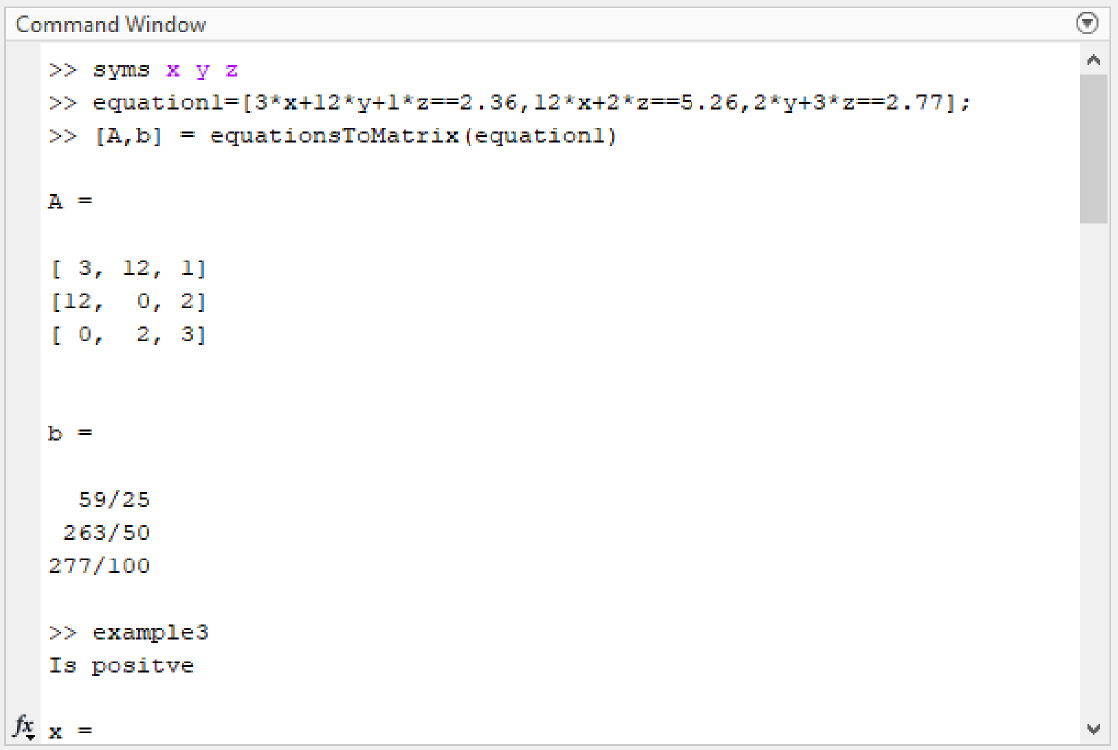
4.6 Exercise (Page 42)

1. Liz buys three apples, a dozen bananas, and one cantaloupe for $2.36. Bob buys a dozen apples and two cantaloupe for $5.26. Carol buys two bananas and three cantaloupe for $2.77. How much do single pieces of each fruit cost?

Solution

Let: Apple = x, Banana = y and Cantaloupe = z

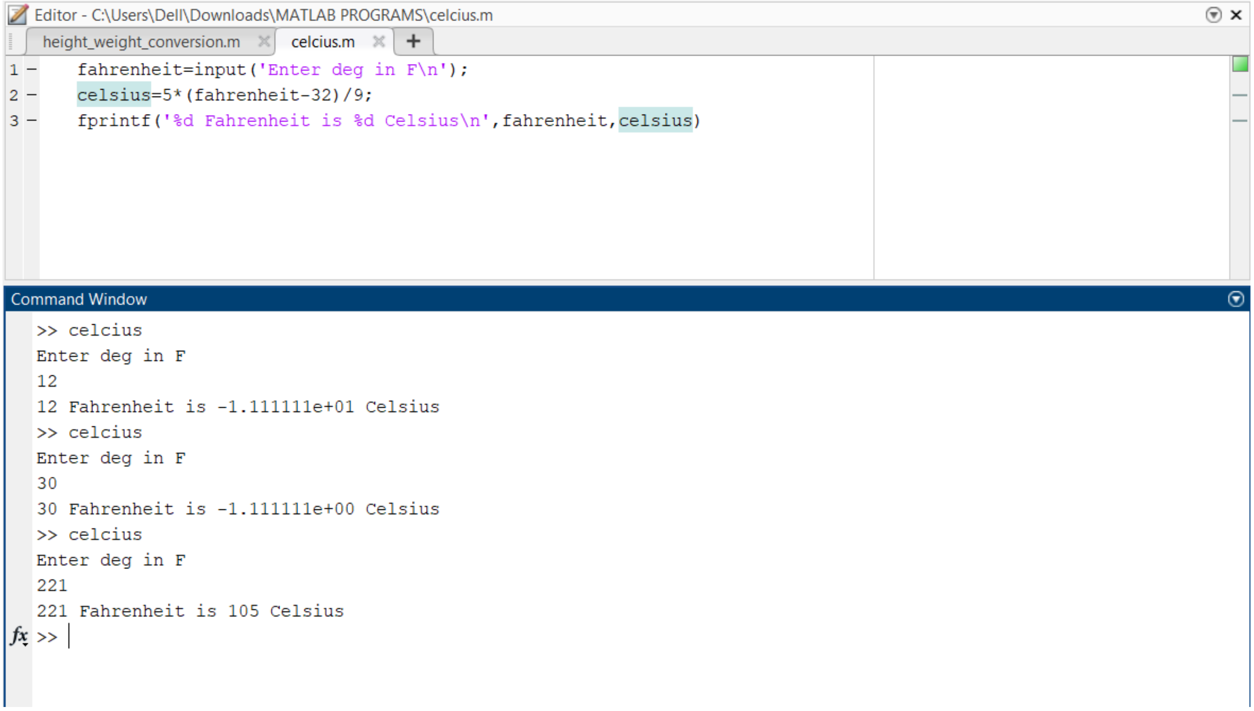
Note: A dozen is a group or set of twelve



```
>> syms x y z
>> equation1=[3*x+12*y+1*z==2.36,12*x+2*z==5.26,2*y+3*z==2.77];
>> [A,b] = equationsToMatrix(equation1)

A =

[ 3, 12, 1]
[12,  0, 2]
[ 0,  2, 3]


b =

  59/25
 263/50
277/100

>> example3
Is positve

x =
```

2. Write a function file that converts temperature in degrees Fahrenheit (◦F) to degrees Centigrade (◦C). Use input and fprintf commands to display a mix of text and numbers. Recall the conversion formulation, C = 5/9 ∗ (F − 32).

Solutions


Editor - C:\Users\Dell\Downloads\MATLAB PROGRAMS\celcius.m
height_weight_conversion.m
celcius.m
Command Window

```
fahrenheit=input('Enter deg in F\n');
celsius=5*(fahrenheit-32)/9;
fprintf('%d Fahrenheit is %d Celsius\n',fahrenheit,celsius)
```

```
>> celcius
Enter deg in F
12
12 Fahrenheit is -1.111111e+01 Celsius
>> celcius
Enter deg in F
30
30 Fahrenheit is -1.111111e+00 Celsius
>> celcius
Enter deg in F
221
221 Fahrenheit is 105 Celsius
>>
```

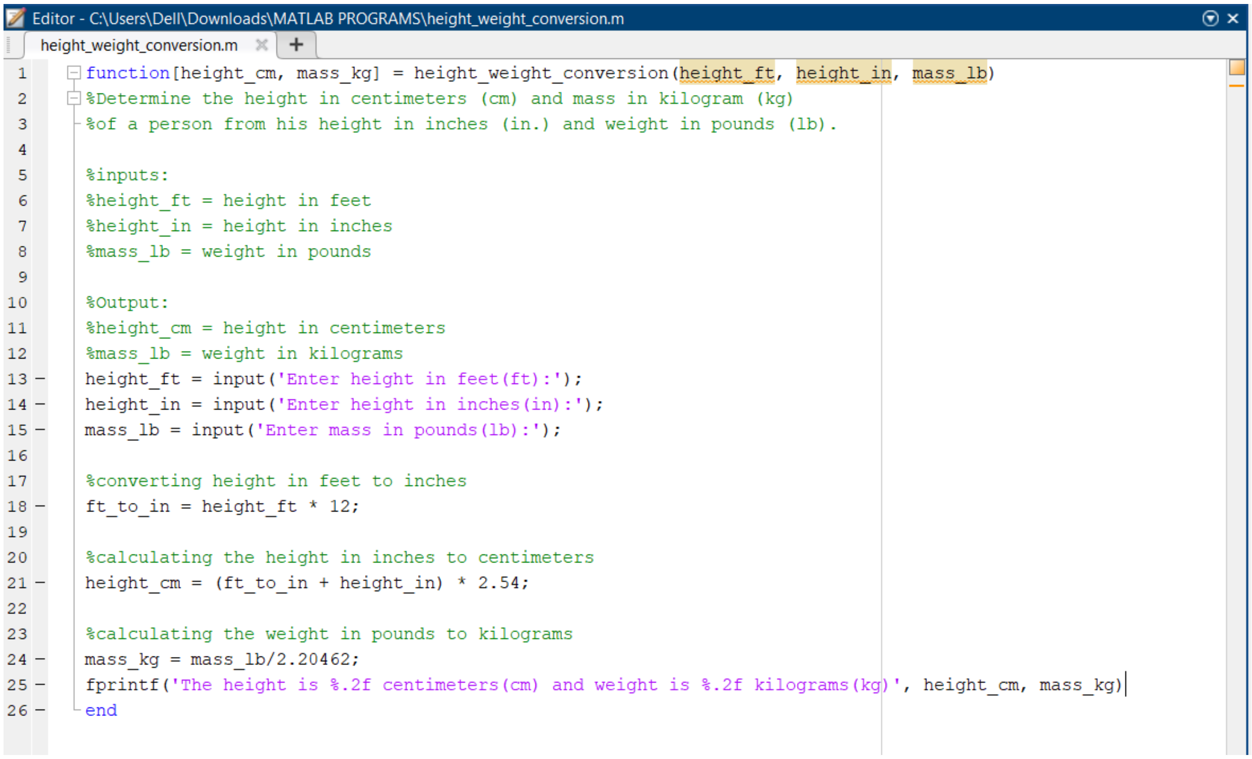
3. Write a user-defined MATLAB function, with two input and two output arguments that determines the height in centimeters (cm) and mass in kilograms (kg)of a person from his height in inches (in.) and weight in pounds (lb).

(a) Determine in SI units the height and mass of a 5 ft.15 in. person who weight 180 lb.

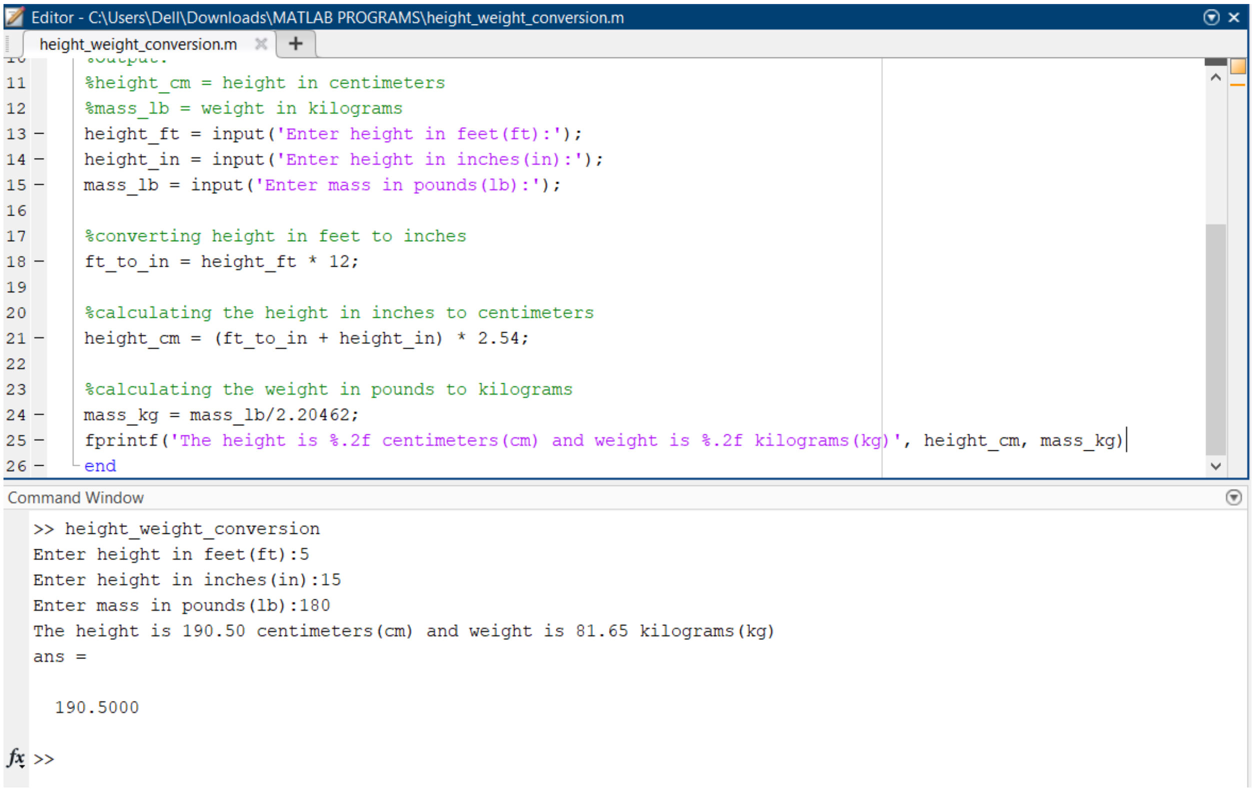
(b) Determine your own height and weight in SI units

Solution

(a) The height and mass of a person with 5ft, 15in and 180lb


Editor - C:\Users\Dell\Downloads\MATLAB PROGRAMS\height_weight_conversion.m
height_weight_conversion.m

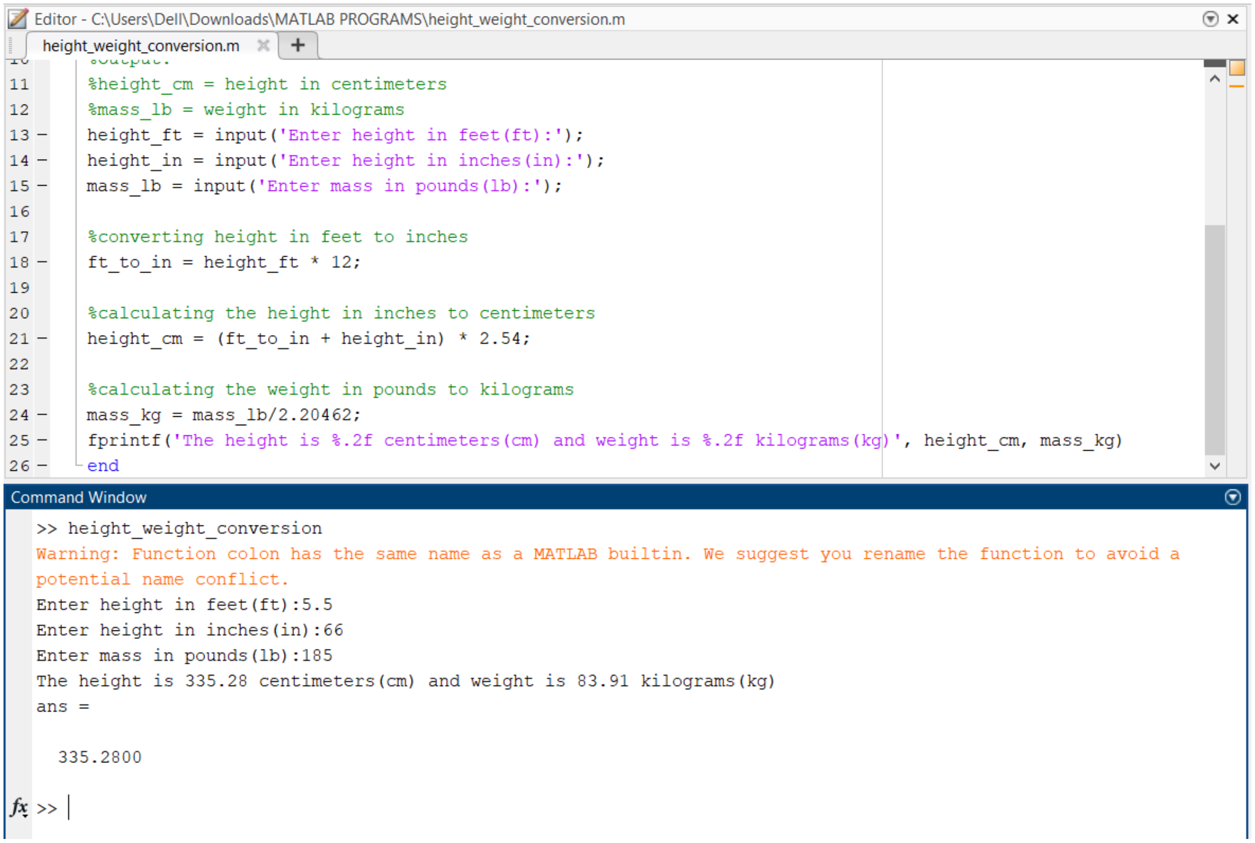
```
function[height_cm, mass_kg] = height_weight_conversion(height_ft, height_in, mass_lb)
%Determine the height in centimeters (cm) and mass in kilogram (kg)
%of a person from his height in inches (in.) and weight in pounds (lb).

%inputs:
%height_ft = height in feet
%height_in = height in inches
%mass_lb = weight in pounds

%Output:
%height_cm = height in centimeters
%mass_lb = weight in kilograms
height_ft = input('Enter height in feet(ft):');
height_in = input('Enter height in inches(in):');
mass_lb = input('Enter mass in pounds(lb):');

%converting height in feet to inches
ft_to_in = height_ft * 12;

%calculating the height in inches to centimeters
height_cm = (ft_to_in + height_in) * 2.54;

%calculating the weight in pounds to kilograms
mass_kg = mass_lb/2.20462;
fprintf('The height is %.2f centimeters(cm) and weight is %.2f kilograms(kg)', height_cm, mass_kg)
end
```


Editor - C:\Users\Dell\Downloads\MATLAB PROGRAMS\height_weight_conversion.m
height_weight_conversion.m
Command Window

```
%height_cm = height in centimeters
%mass_lb = weight in kilograms
height_ft = input('Enter height in feet(ft):');
height_in = input('Enter height in inches(in):');
mass_lb = input('Enter mass in pounds(lb):');

%converting height in feet to inches
ft_to_in = height_ft * 12;

%calculating the height in inches to centimeters
height_cm = (ft_to_in + height_in) * 2.54;

%calculating the weight in pounds to kilograms
mass_kg = mass_lb/2.20462;
fprintf('The height is %.2f centimeters(cm) and weight is %.2f kilograms(kg)', height_cm, mass_kg)
end
```

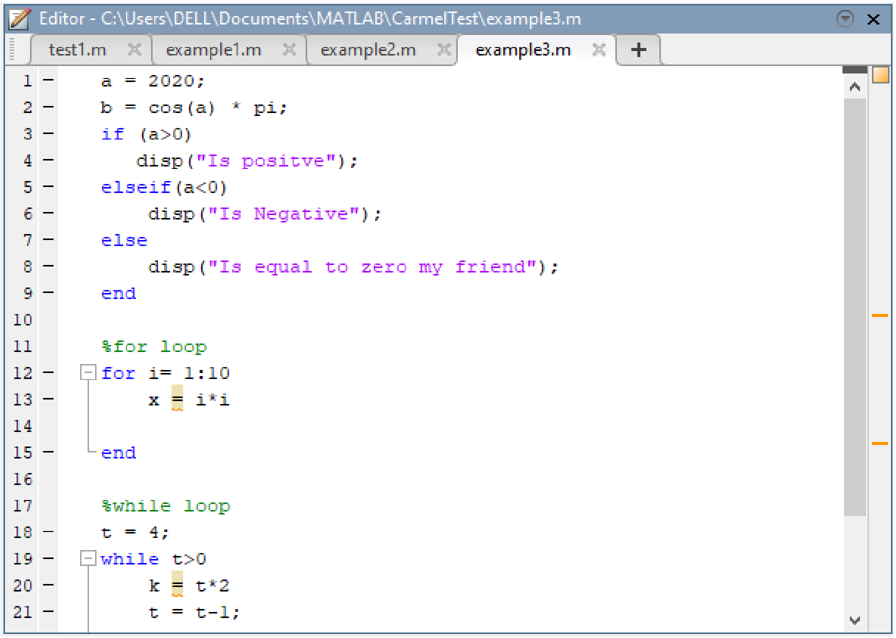
```
>> height_weight_conversion
Enter height in feet(ft):5
Enter height in inches(in):15
Enter mass in pounds(lb):180
The height is 190.50 centimeters(cm) and weight is 81.65 kilograms(kg)
ans =

  190.5000

fx >>
```

(b) My own height and weight will be

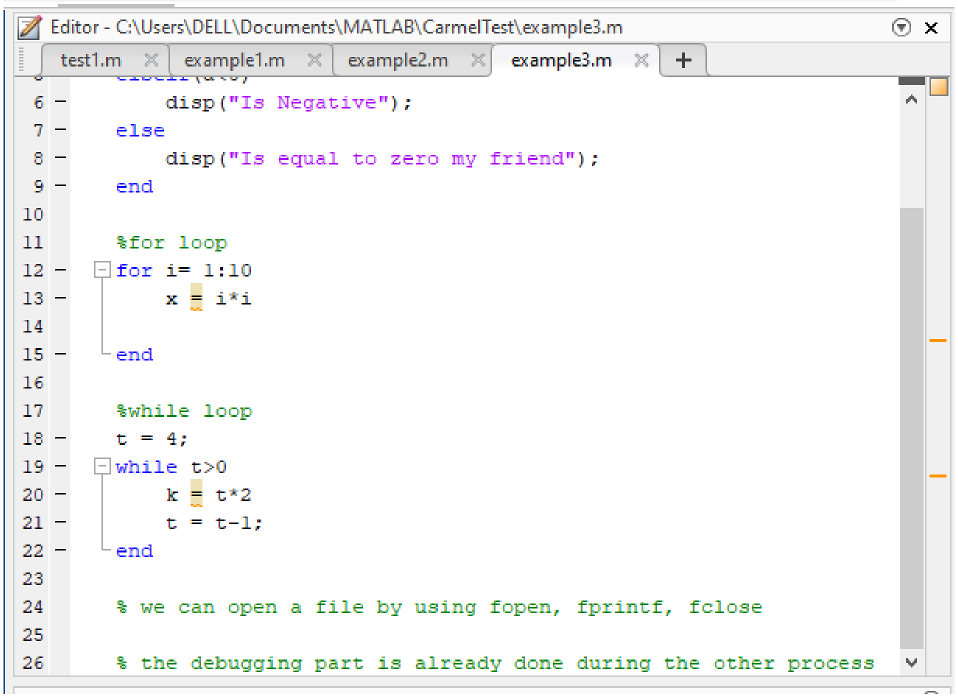

Editor - C:\Users\Dell\Downloads\MATLAB PROGRAMS\height_weight_conversion.m
height_weight_conversion.m
Command Window

```
    %output:
    %height_cm = height in centimeters
    %mass_lb = weight in kilograms
    height_ft = input('Enter height in feet(ft):');
    height_in = input('Enter height in inches(in):');
    mass_lb = input('Enter mass in pounds(lb):');

    %converting height in feet to inches
    ft_to_in = height_ft * 12;

    %calculating the height in inches to centimeters
    height_cm = (ft_to_in + height_in) * 2.54;

    %calculating the weight in pounds to kilograms
    mass_kg = mass_lb/2.20462;
    fprintf('The height is %.2f centimeters(cm) and weight is %.2f kilograms(kg)', height_cm, mass_kg)
end
```

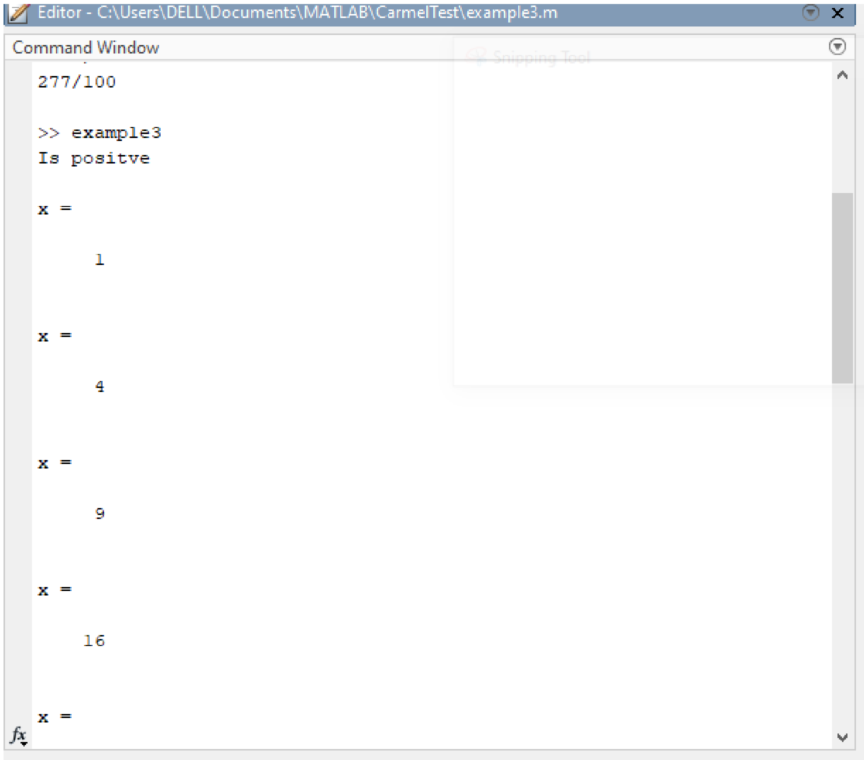
```
>> height_weight_conversion
Warning: Function colon has the same name as a MATLAB builtin. We suggest you rename the function to avoid a
potential name conflict.
Enter height in feet(ft):5.5
Enter height in inches(in):66
Enter mass in pounds(lb):185
The height is 335.28 centimeters(cm) and weight is 83.91 kilograms(kg)
ans =

  335.2800

fx >>
```


Editor - C:\Users\DELL\Documents\MATLAB\CarmelTest\example3.m
test1.m
example1.m
example2.m
example3.m
+

```
a = 2020;
b = cos(a) * pi;
if (a>0)
   disp("Is positve");
elseif(a<0)
    disp("Is Negative");
else
    disp("Is equal to zero my friend");
end

%for loop
for i= 1:10
    x = i*i

end

%while loop
t = 4;
while t>0
    k = t*2
    t = t-1;
```

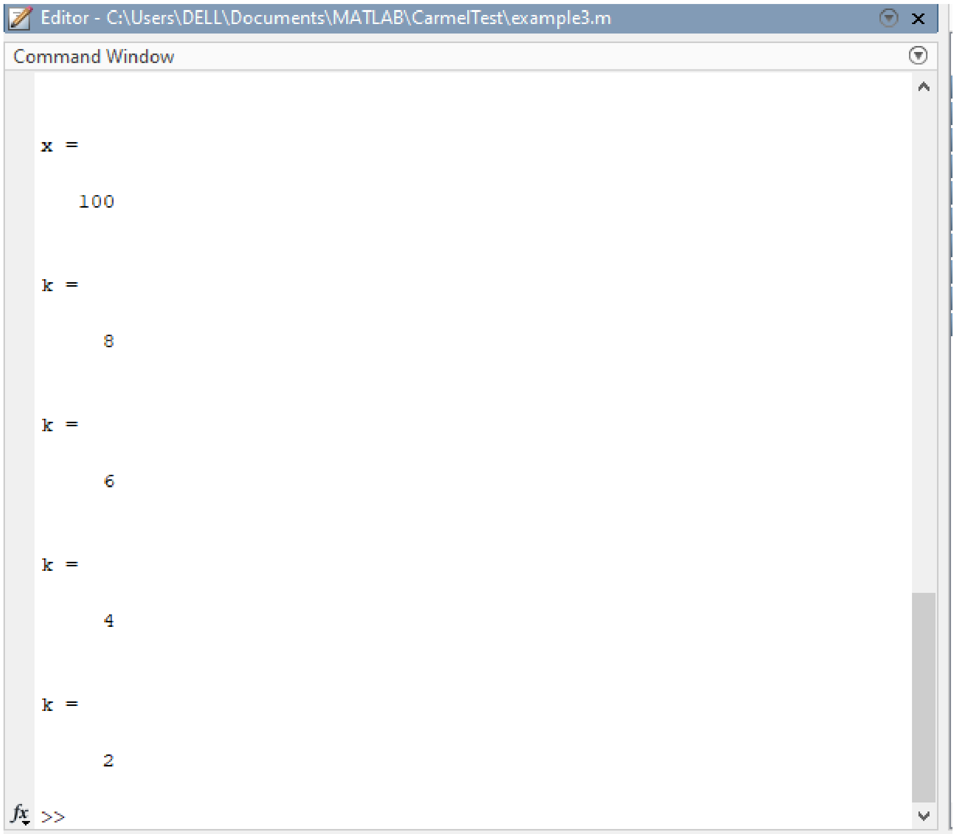

Editor - C:\Users\DELL\Documents\MATLAB\CarmelTest\example3.m
test1.m
example1.m
example2.m
example3.m
+

```
elseif(a<0)
    disp("Is Negative");
else
    disp("Is equal to zero my friend");
end

%for loop
for i= 1:10
    x = i*i

end

%while loop
t = 4;
while t>0
    k = t*2
    t = t-1;
end

% we can open a file by using fopen, fprintf, fclose

% the debugging part is already done during the other process
```



Editor - C:\Users\DELL\Documents\MATLAB\CarmelTest\example3.m
Command Window

```
277/100

>> example3
Is positve

x =

     1


x =

     4


x =

     9


x =

    16


x =
```



Editor - C:\Users\DELL\Documents\MATLAB\CarmelTest\example3.m
Command Window

```

x =

   100


k =

     8


k =

     6


k =

     4


k =

     2

>>
```