

THE NELSON MANDELA AFRICAN INSTITUTION OF SCIENCE AND TECHNOLOGY (NM-AIST)



COURSE TITLE: EMoS 6222, MOBILE TELECOMMUNICATION AND TECHNOLOGY

LECTURER: Dr.Ally Mussa Dida

STUDENT: Salama Ndayisaba

Specialization: Embedded Sytems

Reg. Number: M083/BI19

ASSIGNMENT TITLE: Captures of output of Matlab codes

Date of submission: September 27, 2022

Table of Content

Capture_1: Exercise for entering a vector	2
Capture_2: Exercise for entering a vector	3
Capture_3: Exercise for entering Matrix	4
Capture_4: Exercise for Linear spacing	5
Capture_5: Linear Spacing continuous exercise	6
Capture_6: Exercise of colon operator in matrix	7
Capture_7: Exercise for colon operator in matrix	8
Capture_8: Exercise for a sub-matrix	9
Capture_9: The same exercise	10
Capture_10: The same exercise	11
Capture_11: Exercise for deleting a row or column.....	12
Capture_12: Exercise for dimension	13
Capture_13: Exercise for matrix generators	14
Capture_14: The exercise continues	15
Capture_15: continuous exercise.....	16
Capture_16: Exercise for solving linear equations.....	17
Capture_17: continuous exercise.....	18
Capture_18: Exercise for factorial and format	19

Capture_1: Exercise for entering a vector

```
Command Window
>> 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
w-v'=w
↑
Incorrect use of '=' operator. Assign a value to a variable using '=' and compare values for
equality using '=='.

>> w-v'

ans =

     0
     0
```

Capture_2: Exercise for entering a vector

```
Command Window
>> w=v'

ans =

     0
     0
     0
     0
     0

>> v(1:3)

ans =

     1     4     7

>> v(3,end)
Index in position 1 exceeds array bounds (must not exceed 1).

>> w = v'

w =

     1
     4
     7
    10
    13
```

Capture_3: Exercise for entering Matrix

```
Command Window
// v(1:3)
Unrecognized function or variable 'v'.

>> w(1:3)
Unrecognized function or variable 'w'.

>> 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

>> A(3,3) = 0

A =

     1     2     3
     4     5     6
     7     8     0
```

f_x >> |

Capture_4: Exercise for Linear spacing

```
Command Window
>> 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

Columns 19 through 27
    1.1310    1.1938    1.2566    1.3195    1.3823    1.4451    1.5080    1.5708    1.6336

Columns 28 through 36
    1.6965    1.7593    1.8221    1.8850    1.9478    2.0106    2.0735    2.1363    2.1991

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
```

Capture_5: Linear Spacing continuous exercise

Command Window									
Columns 40 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	
Columns 91 through 99									
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								
fx >>									

Capture_6: Exercise of colon operator in matrix

```
Command Window

>> A(2,:)

ans =

     4     5     6

>> A(:,2:3)

ans =

     2     3
     5     6
     8     0

>> A(:,2)=[]

A =

     1     3
     4     6
     7     0

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

B =

     4     6
     7     0
```


Capture_7: Exercise for colon operator in matrix

```
Command Window

    2     3
    5     6
    8     0

>> A(:,2)=[]

A =

    1     3
    4     6
    7     0

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

B =

    4     6
    7     0

>> C = A([2 1 3],:)

C =

    4     6
    1     3
    7     0

fx >> |
```

Capture_8: Exercise for a sub-matrix

```
Command Window

>> C = A([2 1 3], :)

C =

     4     6
     1     3
     7     0

>> A(:)

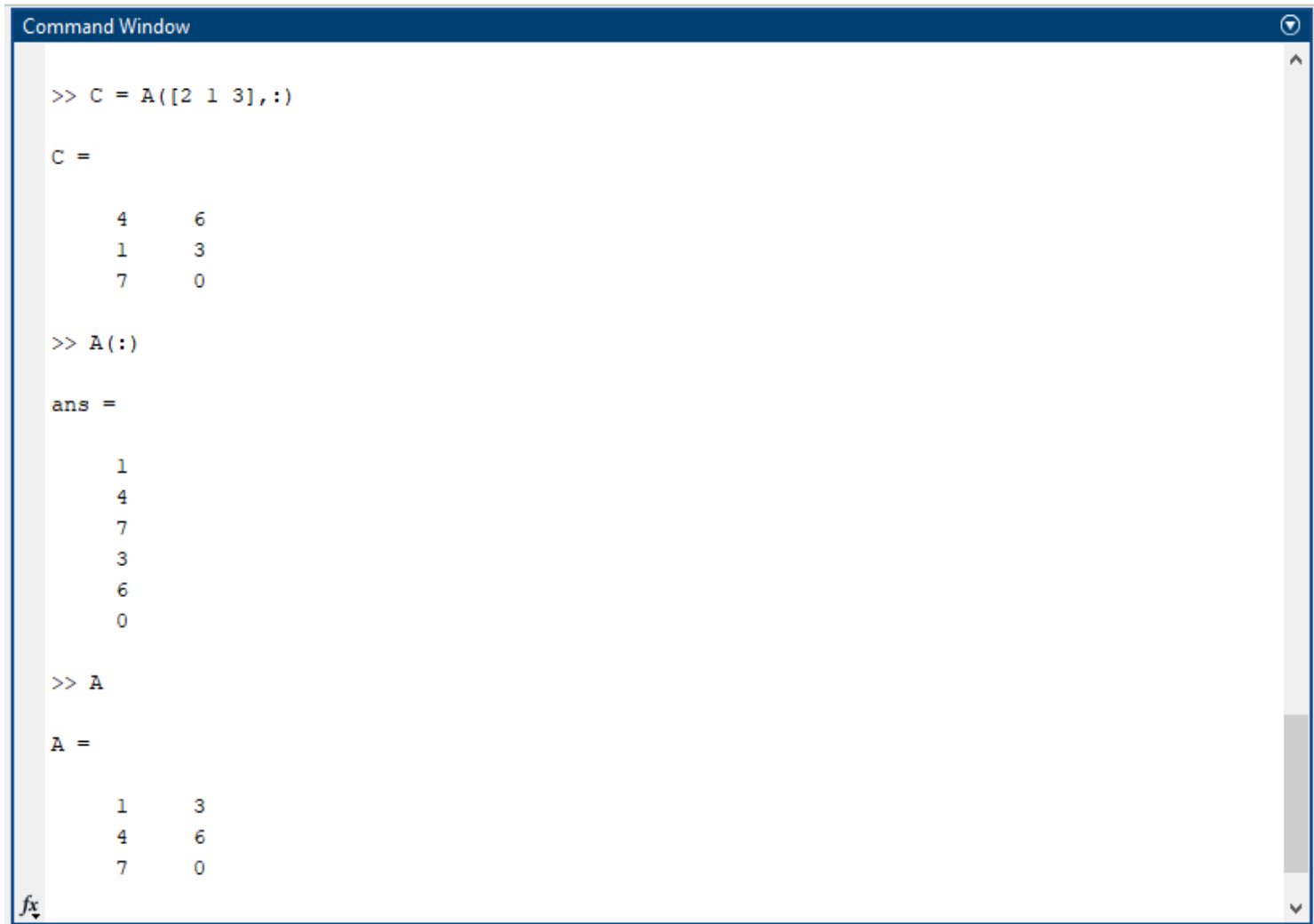
ans =

     1
     4
     7
     3
     6
     0

>> A

A =

     1     3
     4     6
     7     0
```

A screenshot of the MATLAB Command Window. The window has a dark blue title bar with the text "Command Window" and a small icon on the right. The main area is white with black text. The text shows a sequence of commands and their outputs. The first command is "C = A([2 1 3], :)", which outputs a 3x2 matrix C. The second command is "A(:)", which outputs a column vector of all elements of A. The third command is "A", which outputs the original 3x2 matrix A. The window has a vertical scrollbar on the right and a small icon in the bottom left corner.

Capture_9: The same exercise

```
Command Window

>> A

A =

     1     3
     4     6
     7     0

>> A(2:3,2:3)
Index in position 2 exceeds array bounds (must not exceed 2).

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

ans =

     0
     6
     3

>> A([1 3],[2 3])
Index in position 2 exceeds array bounds (must not exceed 2).

>> A(3,:) = []

A =

     1     3
     4     6
```

Capture_10: The same exercise

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

ans =

    0
    6
    3

>> A([1 3],[2 3])
Index in position 2 exceeds array bounds (must not exceed 2).

>> A(3,:) = []

A =


    1     3
    4     6

>> A = [A(1,:);A(2,:);[7 8 0]]
Error using vertcat
Dimensions of arrays being concatenated are not consistent.

>> size(A)

ans =

    2     2
```



Capture_11: Exercise for deleting a row or column

```
Command Window

>> A(3,:) = []

A =

     1     3
     4     6

>> A = [A(1,:);A(2,:);[7 8 0]]
Error using vertcat
Dimensions of arrays being concatenated are not consistent.

>> size(A)

ans =

     2     2

>> [m,n]=size(A)

m =

     2

n =

     2
```

Capture_12: Exercise for dimension

```
Command Window

>> size(A)

ans =

     2     2

>> [m,n]=size(A)

m =

     2

n =

     2

>> A'

ans =

     1     4
     3     6

>> 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]]
      ↑
```

Capture_13: Exercise for matrix generators

```
Command Window

>> B = [A 10*A; -A [1 0 0; 0 1 0; 0 0 1]]
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> b=ones(3,1)

b =

     1
     1
     1

>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1

>> c=zeros(2,3)

c =

     0     0     0
     0     0     0

fx >> |
```

Capture_14: The exercise continues

```
Command Window

>> B = [A 10*A; -A [1 0 0; 0 1 0; 0 0 1]]
Error using horzcat
Dimensions of arrays being concatenated are not consistent.

>> b=ones(3,1)

b =

     1
     1
     1

>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1

>> c=zeros(2,3)

c =

     0     0     0
     0     0     0

fx >> D = [C zeros(2) * ones(2) eyes(2)]
```


Capture_15: continuous exercise

```
Command Window
0    0    1

>> c=zeros(2,3)

c =

    0    0    0
    0    0    0

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

>> C = A.*B

C =

     4    18
    28     0

>> A.^2

ans =

     1     9
    16    36

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

Capture_16: Exercise for solving linear equations

```
Command Window

x =

    0
   -1
    1

>> A=[1 2 3; 4 5 6; 7 8 0];
>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = inv(A)*b

x =

   -1.0000
    1.0000
   -0.0000

>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = A\b

x =

   -1.0000
    1.0000
   -0.0000

fx >> |
```

Capture_17: continuous exercise

```
Command Window
1.0000
-0.0000

>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = A\b

x =

-1.0000
1.0000
-0.0000

>> A = [1 2 3; 4 5 6; 7 8 0];
>> 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

fx >> |
```

Capture_18: Exercise for factorial and format

```
Command Window
>> edit
>> f =factorial(5)

f =

    120

>> format short eng
>> pi

ans =

    3.1416e+000

>> format long eng
>> pi

ans =

    3.14159265358979e+000

>> help function_name>subfunction_name

function_name>subfunction_name not found.

Use the Help browser search field to search the documentation, or
fx type "help help" for help command options, such as help for methods.
```