Exercise 1

Vector Operations

Q1. Write a program to create two row vectors of 10 elements each and perform the arithmetic operation as mentioned below:

1. **Addition**
2. **Subtraction**
3. **Multiplication**
4. **Append**
5. **Transpose**
6. **Power**

Logic:

Step 1: First we have to create a new script that is named as demo.m.

Step 2: Now we have to initialise two row vectors named as A and B.

Step 3: X and Y are row vectors.

Step 4: We have to use the arithmetic symbols '+', '-', '.*' for addition, subtraction and multiplication.

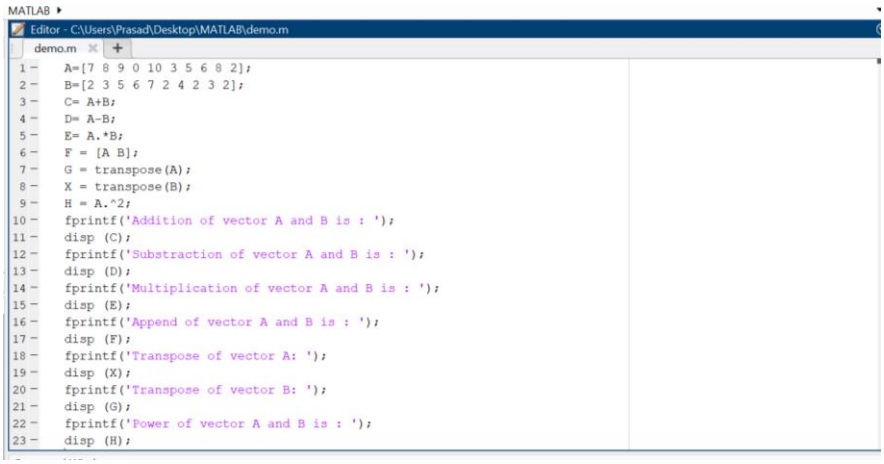
Step 5: We have to calculate power using ^ symbol.

Step 6: The results are printed using disp( ) function.

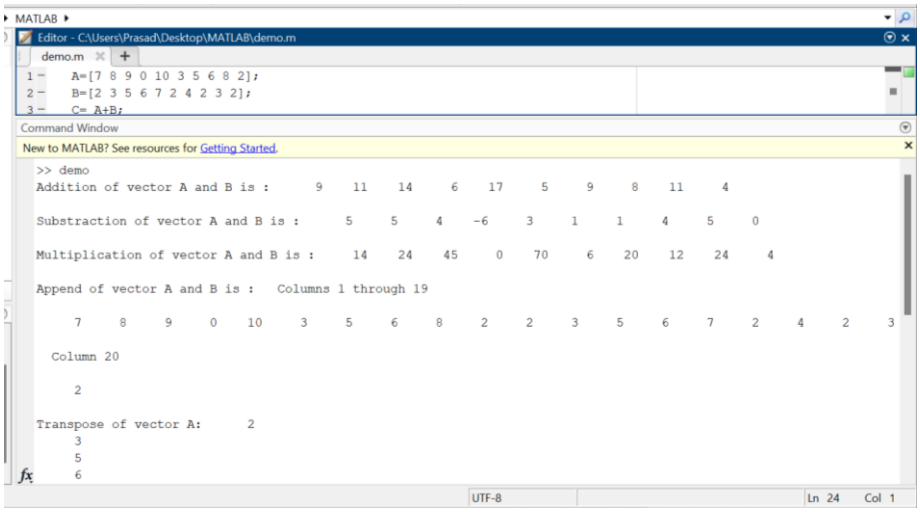
Step 7: The result would be printed in the command window.

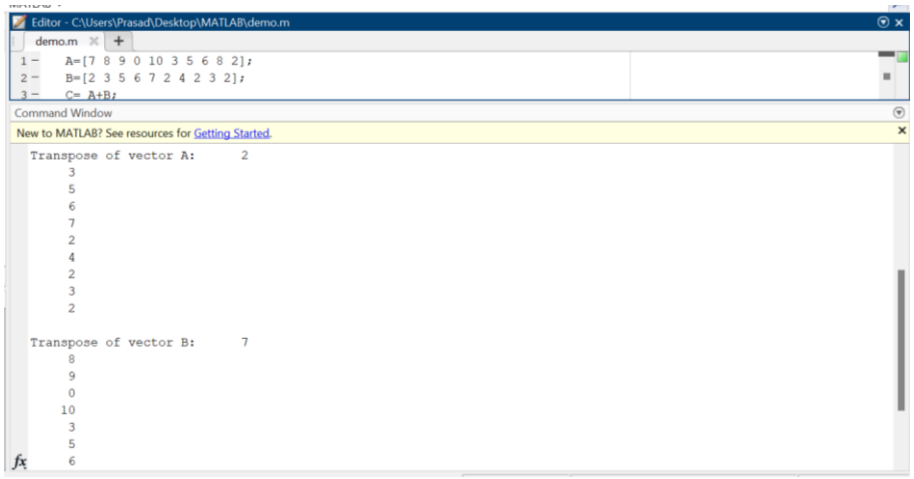
Step 8: Stop

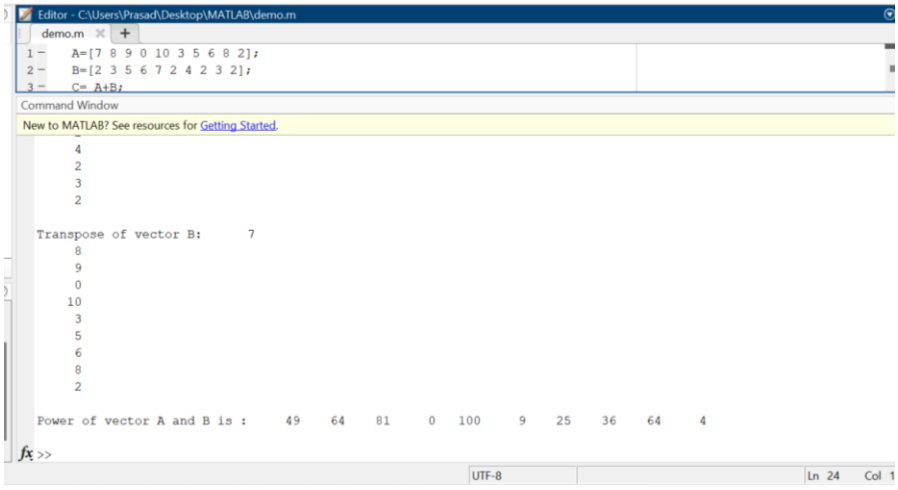
Code:



Output:


MATLAB
Editor - C:\Users\Prasad\Desktop\MATLAB\demo.m
demo.m
1 - A=[7 8 9 0 10 3 5 6 8 2];
2 - B=[2 3 5 6 7 2 4 2 3 2];
3 - C= A+B;
Command Window
New to MATLAB? See resources for Getting Started.
>> demo
Addition of vector A and B is : 9 11 14 6 17 5 9 8 11 4
Substraction of vector A and B is : 5 5 4 -6 3 1 1 4 5 0
Multiplication of vector A and B is : 14 24 45 0 70 6 20 12 24 4
Append of vector A and B is : Columns 1 through 19
7 8 9 0 10 3 5 6 8 2 2 3 5 6 7 2 4 2 3
Column 20
2
Transpose of vector A: 2
3
5
6
UTF-8
Ln 24 Col 1


Editor - C:\Users\Prasad\Desktop\MATLAB\demo.m
demo.m
1 - A=[7 8 9 0 10 3 5 6 8 2];
2 - B=[2 3 5 6 7 2 4 2 3 2];
3 - C= A+B;
Command Window
New to MATLAB? See resources for Getting Started.
Transpose of vector A: 2
3
5
6
7
2
4
2
3
2
Transpose of vector B: 7
8
9
0
10
3
5
6


Editor - C:\Users\Prasad\Desktop\MATLAB\demo.m
demo.m
1 - A=[7 8 9 0 10 3 5 6 8 2];
2 - B=[2 3 5 6 7 2 4 2 3 2];
3 - C= A+B;
Command Window
New to MATLAB? See resources for Getting Started.
4
2
3
2
Transpose of vector B: 7
8
9
0
10
3
5
6
8
2
Power of vector A and B is : 49 64 81 0 100 9 25 36 64 4
>>
UTF-8
Ln 24 Col 1

Q2. Write a program to create two column vectors of 10 elements each and perform the arithmetic operation as mentioned below:

1. Addition

2. Subtraction

3. Multiplication

4. Append

5. Transpose

6. Power

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m.

Step 3: Now just create column vectors with the help of semi colon between the elements.

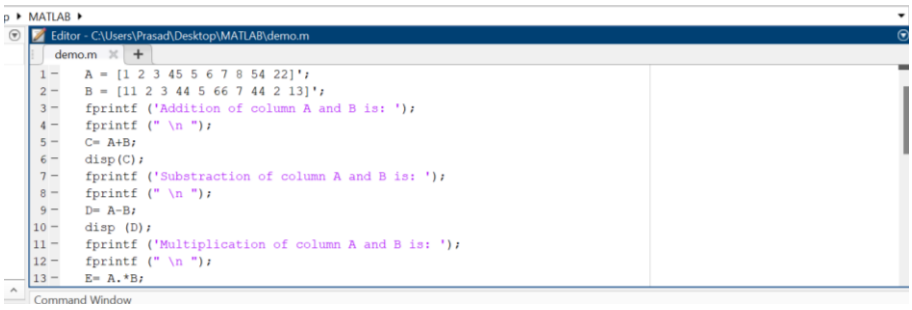
Step 4: So, we have to just use the arithmetic symbols '+', '-', '.*' for addition, subtraction and multiplication.

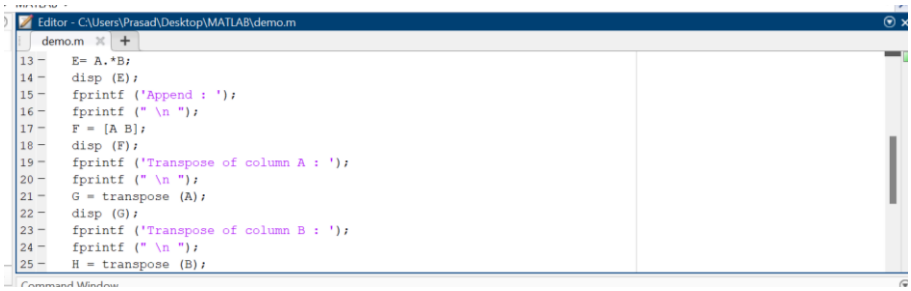
Step 5: We have to calculate power using ^ symbol.

Step 6: So, the vectors (A and B) and results are printed using disp( ) function.

Step 7: Stop

Code:





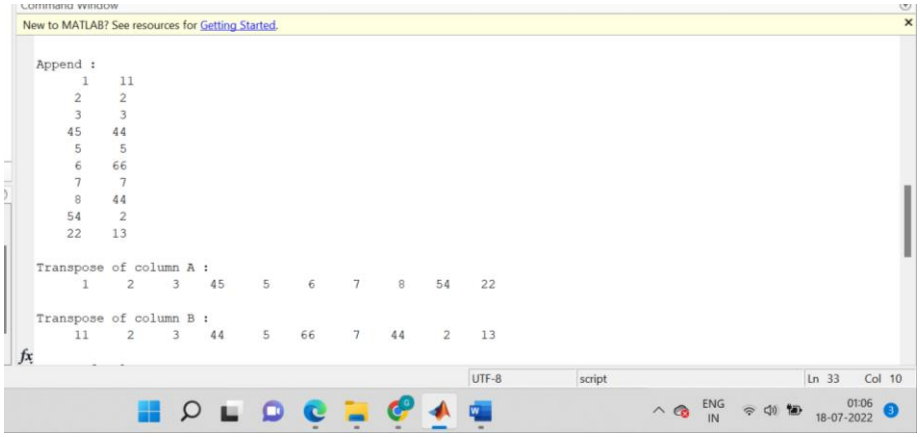
```
fprintf ('Transpose of column B : ');
fprintf (" \n ");
H = transpose (B);
disp (H);
fprintf ('Power of column A : ');
fprintf (" \n ");
I = A.^2;
disp (I);
fprintf ('Power of column B : ');
fprintf (" \n ");
J = B.^2;
disp (J);
```

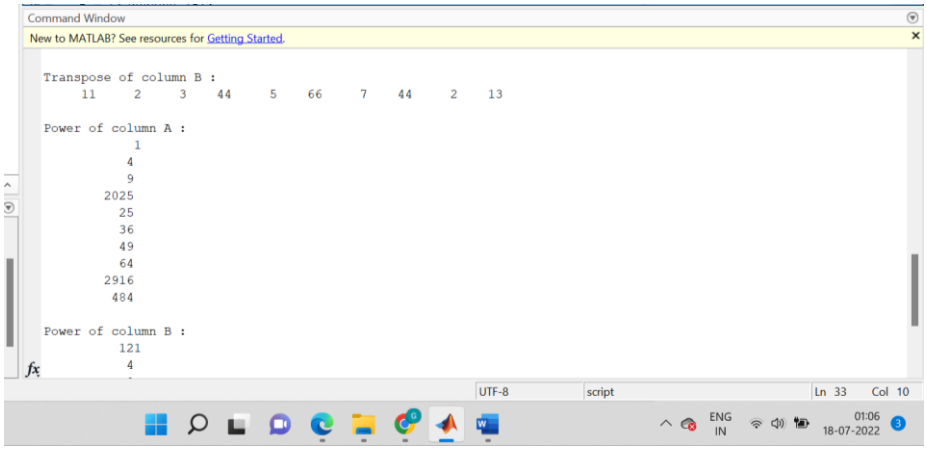
Output:

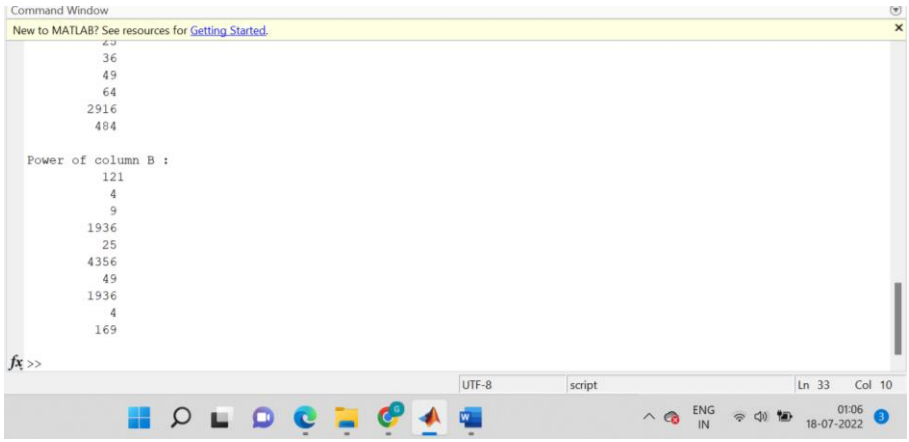
```
>> demo
Addition of column A and B is:
     12
      4
      6
     89
     10
     72
     14
     52
     56
     35

Substraction of column A and B is:
    -10
      0
      0
      1
      0
    -60
```

```
Substraction of column A and B is:
    -10
      0
      0
      1
      0
    -60
      0
    -36
     52
      9

Multiplication of column A and B is:
           11
            4
            9
         1980
           25
```

```
Multiplication of column A and B is:
           11
            4
            9
         1980
           25
          396
           49
          352
          108
          286

Append :
     1    11
     2     2
     3     3
    45    44
     5     5
```


New to MATLAB? See resources for Getting Started.
Append :
1 11
2 2
3 3
45 44
5 5
6 66
7 7
8 44
54 2
22 13
Transpose of column A :
1 2 3 45 5 6 7 8 54 22
Transpose of column B :
11 2 3 44 5 66 7 44 2 13
UTF-8
script
Ln 33 Col 10
ENG
IN
01:06
18-07-2022


Command Window
New to MATLAB? See resources for Getting Started.
Transpose of column B :
11 2 3 44 5 66 7 44 2 13
Power of column A :
1
4
9
2025
25
36
49
64
2916
484
Power of column B :
121
4
UTF-8
script
Ln 33 Col 10
ENG
IN
01:06
18-07-2022


Command Window
New to MATLAB? See resources for Getting Started.
36
49
64
2916
484
Power of column B :
121
4
9
1936
25
4356
49
1936
4
169
>>
UTF-8
script
Ln 33 Col 10
ENG
IN
01:06
18-07-2022

Q3. Write a program to calculate the Square and Cube of any number.

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m

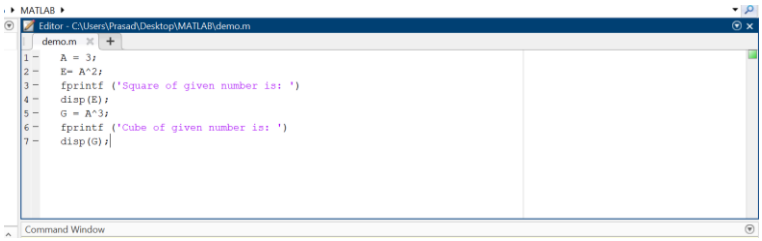
Step 3: Initialise A = 3

Step 4: So, the square and cube of the number are calculated using ^ operator.

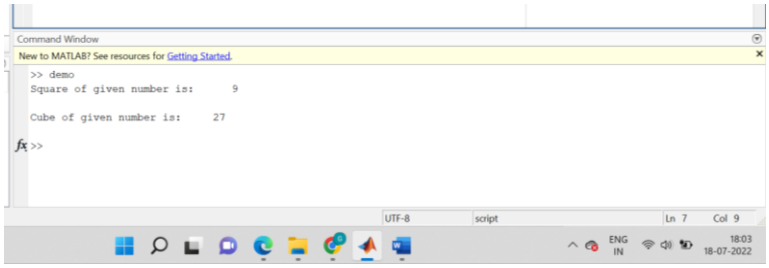
Step 5: The number and results are printed using disp( ) function.

Step 6: Stop

Code:



Output:



Q4. Write a program to calculate the hypotenuse of a right-angle triangle. (Use sqrt() function)

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m.

Step 3: Now we have to just initialise the perpendicular and base of a right-angle triangle as A and B.

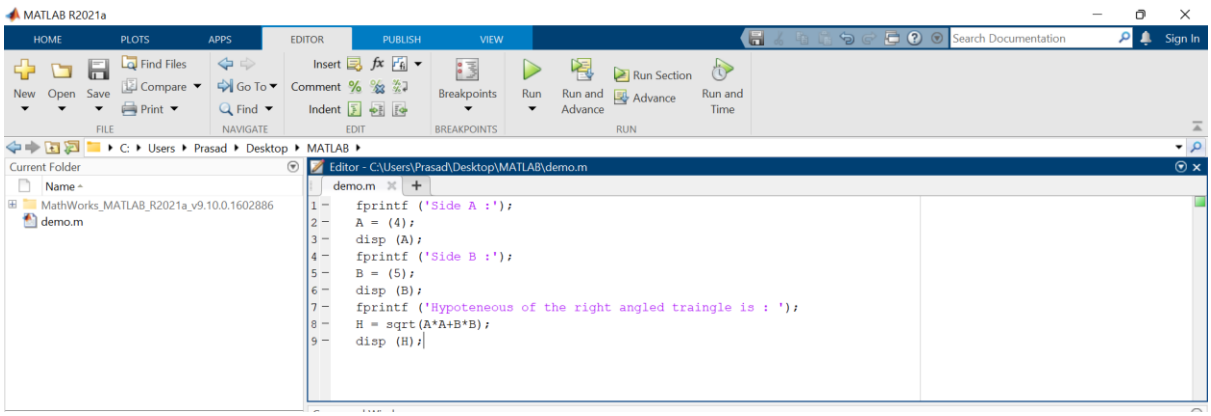
Step 4: So, the square of the A and B operator is used and sqrt( ) are used to calculate the square root of the squared variables and store them as Hypotenuse _H .

Step 5: Also the sides and results are printed using disp( ) function.

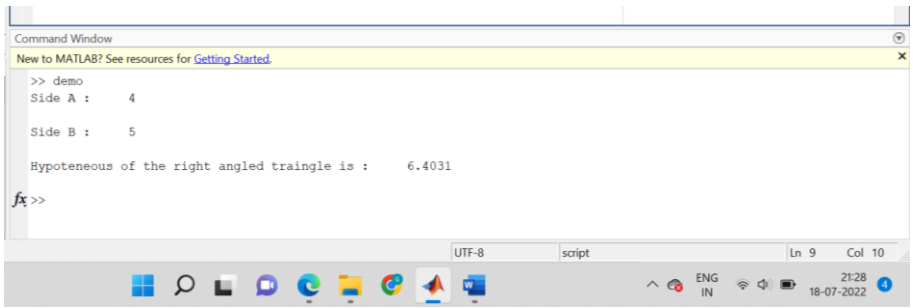
Step 6: Now, the result would be printed in the command window. Comments can be added using % sign

Step 7: Stop

Code:



Output:



Q5. Write a program to calculate the Dot Product of two vectors consisting of six elements each.

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m that would be the MATLAB editor window where the MATLAB programs can be loaded, edited and saved.

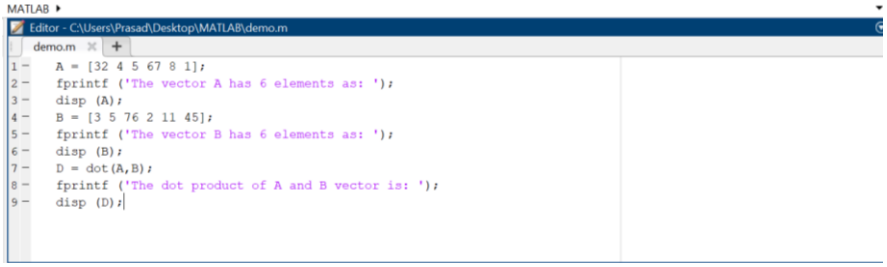
Step 3: Now we have to just initialise two vectors named as A and B each containing six elements.

Step 4: Just calculate the dot product for both vectors using dot( ) function.

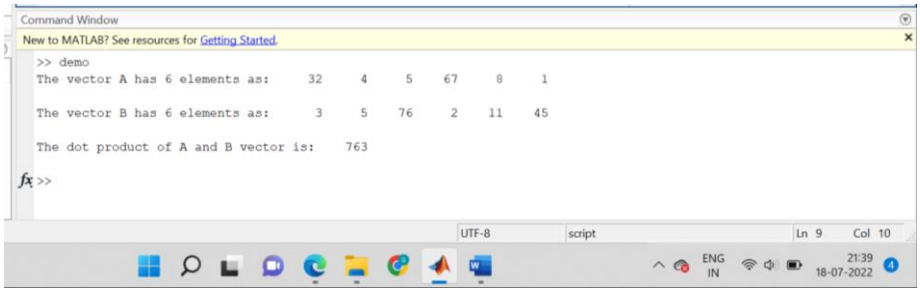
Step 5: The results can be shown using the disp() function.

Step 6: Stop

Code:



Output:



Q6. Write a program to calculate the magnitude of a vector consisting of seven elements.

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m.

Step 3: We have to now take the product of the vector with itself by just using array multiplication.

Step 4: Now take the sum function for sum of squares of elements of vector A.

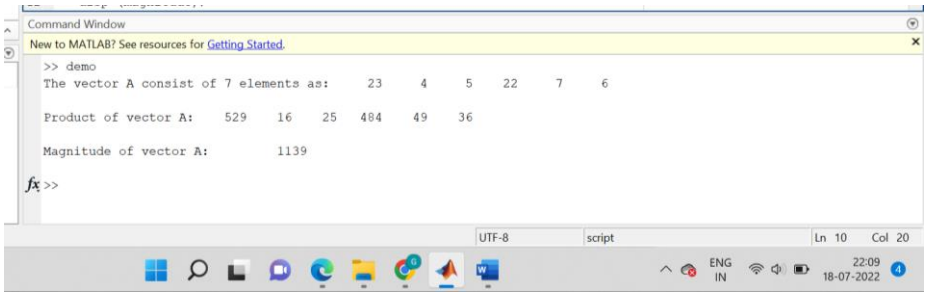
Step 5: So, use the sqrt function to get the square root of the sum which is also the magnitude of the vector A.

Step 6: disp( ) function is used for displaying vector and results.

Step 7: Stop

Code:

Output:



Exercise 2

Matrix Operations

Q1. Write a program to perform the scalar operations on a 4X3 matrix.

Logic:

Step 1: Start

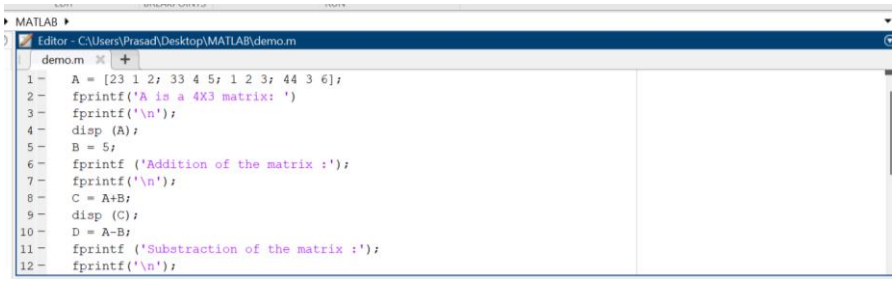
Step 2: First we have to create a new script named as demo.m

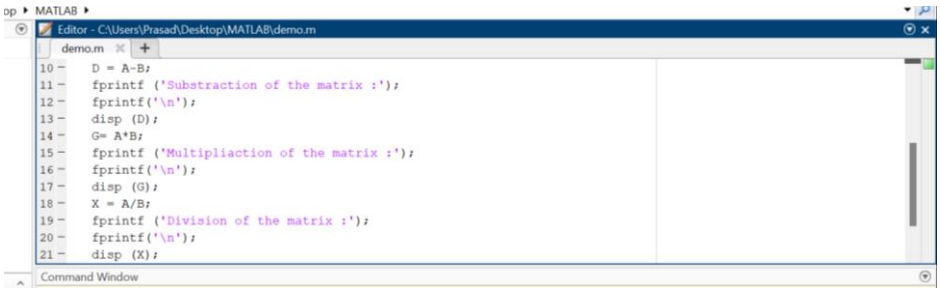
Step 3: Now just create a matrix and initialize a variable.

Step 4: So, the scalar operations are performed on matrix such as- Division is performed using ‘ Operator. Multiplication is performed using ‘*’ operator. Addition is performed using ‘+’ operator. Subtraction is performed using ‘-‘ operator.

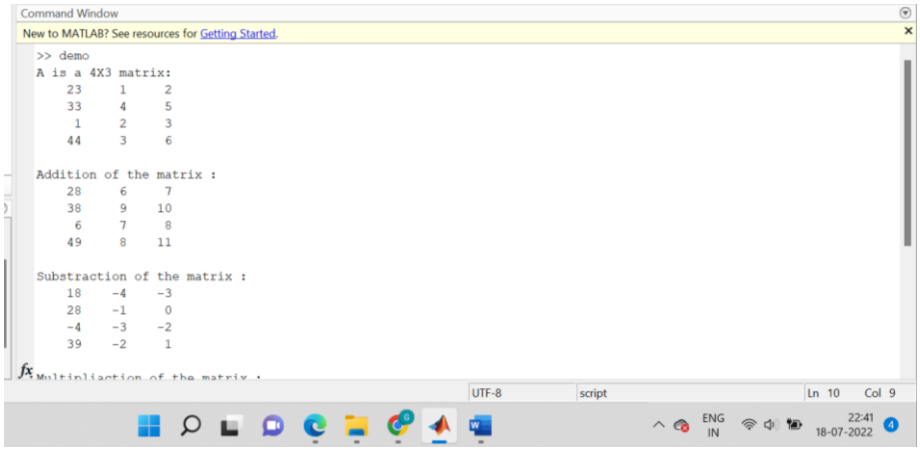
Step 5: Stop

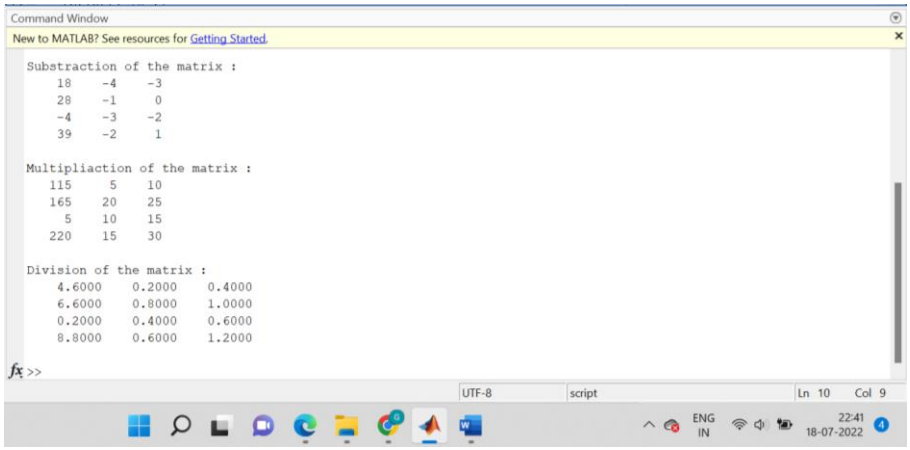
Code:





Output:




Command Window
New to MATLAB? See resources for Getting Started.
Substraction of the matrix :
18 -4 -3
28 -1 0
-4 -3 -2
39 -2 1
Multipliaction of the matrix :
115 5 10
165 20 25
5 10 15
220 15 30
Division of the matrix :
4.6000 0.2000 0.4000
6.6000 0.8000 1.0000
0.2000 0.4000 0.6000
8.8000 0.6000 1.2000
>>
UTF-8
script
Ln 10 Col 9
ENG
IN
22:41
18-07-2022

Q2. Write a program to perform

1. Matrix multiplication (Direct)

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m

Step 3: Now just make 2 matrices.

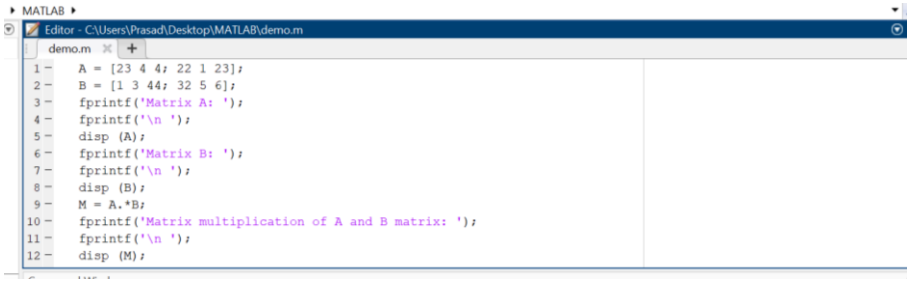
Step 4: We have to use '*' symbol for direct multiplication.

Step 5: The result would be printed in the command window.

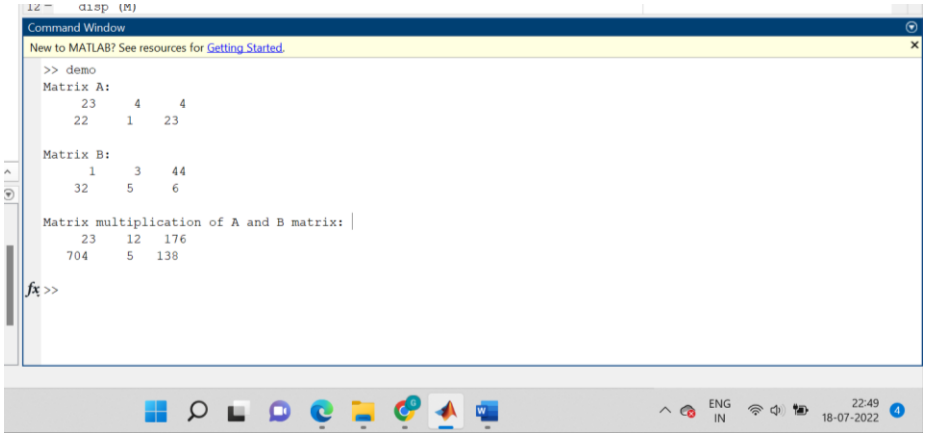
Step 6: Use disp() function for displaying result.

Step 7: Stop

Code:



Output:



2. Prod ()

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m.

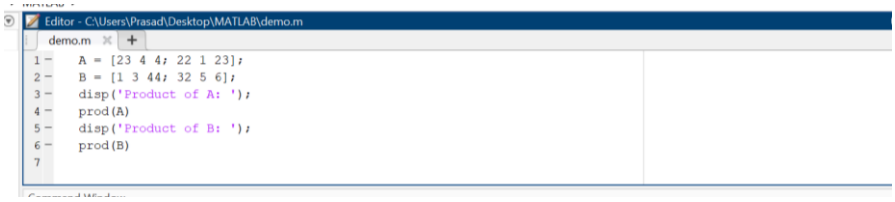
Step 3: Now just create 2 matrices.

Step 4: So, use the Prod() to calculate the product of each column.

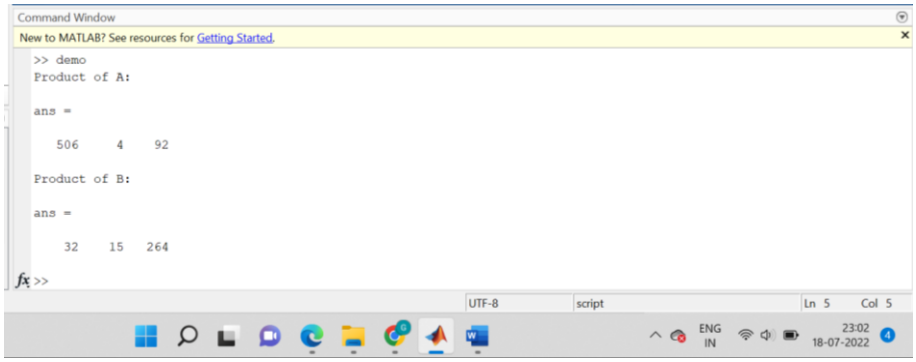
Step 5: Result would be printed in the command window. Use disp() for displaying result.

Step 6: Stop

Code:



Output:



Q3. Write a program to concatenate two 4X4 matrices vertically and horizontally.

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m.

Step 3: Now just create a two 4X4 matrices.

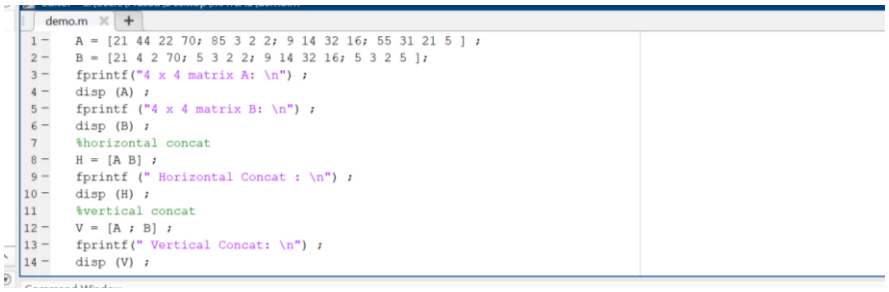
Step 4: So, we have to concatenate horizontally so we have to use square brackets and a comma to separate the variables.

Step 5: Now, to concatenate vertically use square brackets and a semi colon to separate the variables.

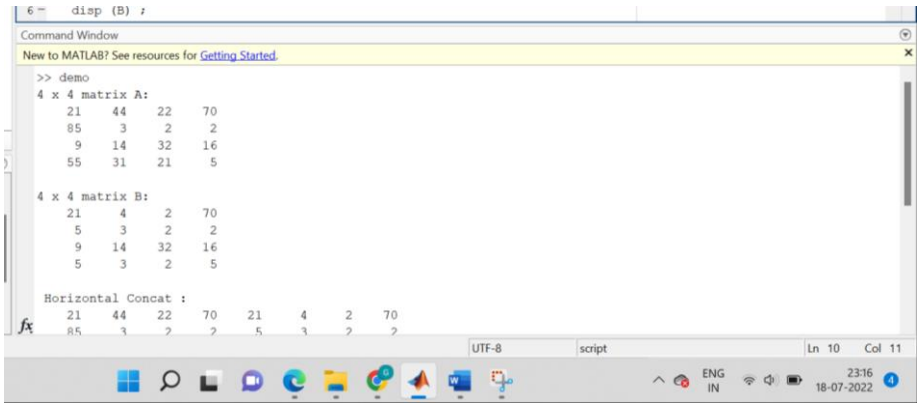
Step 6: We know that the matrix and results are printed using disp( ) function.

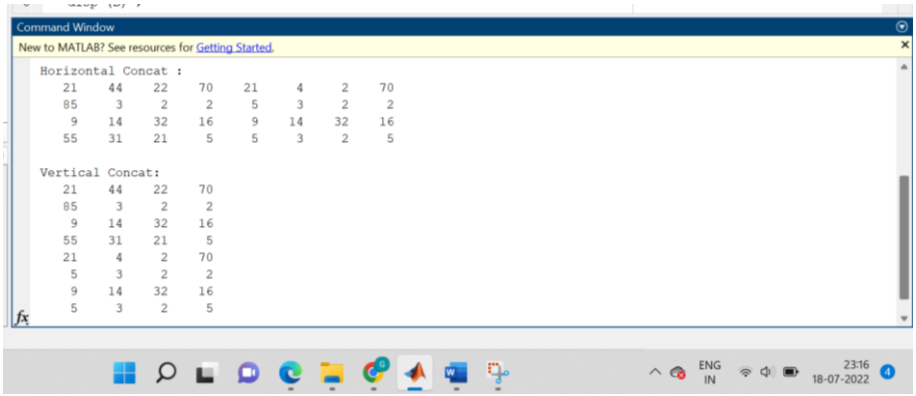
Step 7: Stop

Code:



Output:




Command Window
New to MATLAB? See resources for Getting Started.
Horizontal Concat :
21 44 22 70 21 4 2 70
85 3 2 2 5 3 2 2
9 14 32 16 9 14 32 16
55 31 21 5 5 3 2 5
Vertical Concat:
21 44 22 70
85 3 2 2
9 14 32 16
55 31 21 5
21 4 2 70
5 3 2 2
9 14 32 16
5 3 2 5
ENG
IN
23:16
18-07-2022

Q4. Write a program to calculate the determinant and inverse of the 3X3 matrix.

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m.

Step 3: Now we just have to create a 3 x 3 matrix A.

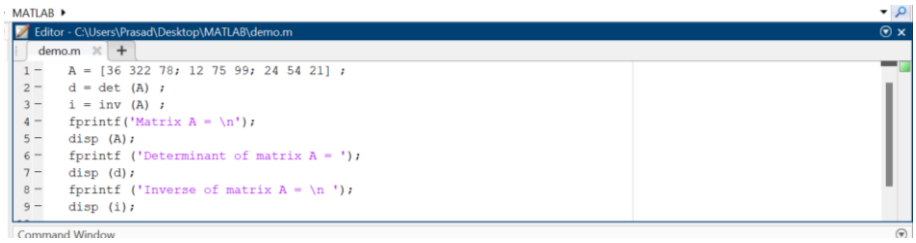
Step 4: The d = det(A) that will return the determinant of square matrix A.

Step 5: We have i = inv(A) that will just compute the inverse of square matrix A.

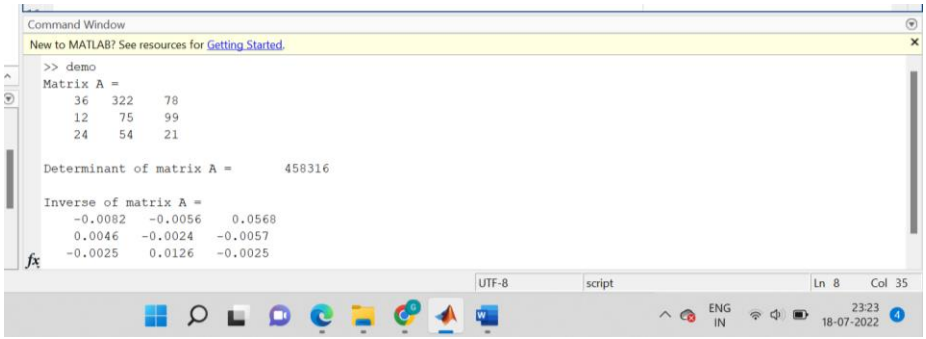
Step 6: We know that the matrix and results are printed using disp( ) function.

Step 7: Stop

Code:



Output:



Exercise 3

Arrays Operations (2-D and Multi-Dimensional)

Q1. Write a program to create an array of size 5X10 to perform the following functions: (Use new variables to store the values for each sub-question).

Logic:

Step 1: Start

Step 2: We have to first create a new script named as demo.m

Step 3: So, first to access the columns from 5 to 10, ':' is used. It's a range operator.

Step 4:Now if we want to access the rows from 2 to 5, ':' is used. It's a range operator.

Step 5: Now if we want to access any one number within an array, specify the row and column.

Step 6: Then to access the last row, enter the row number and just leave the column part empty.

Step 7: If we want to access the 6thcolumn, enter the column number and now just you have to leave the row part empty.

Step 8: Also, if we want to choose the even rows in the array, enter 2:2:end so this simply means that the range will start from index 2 with a gap of 2.

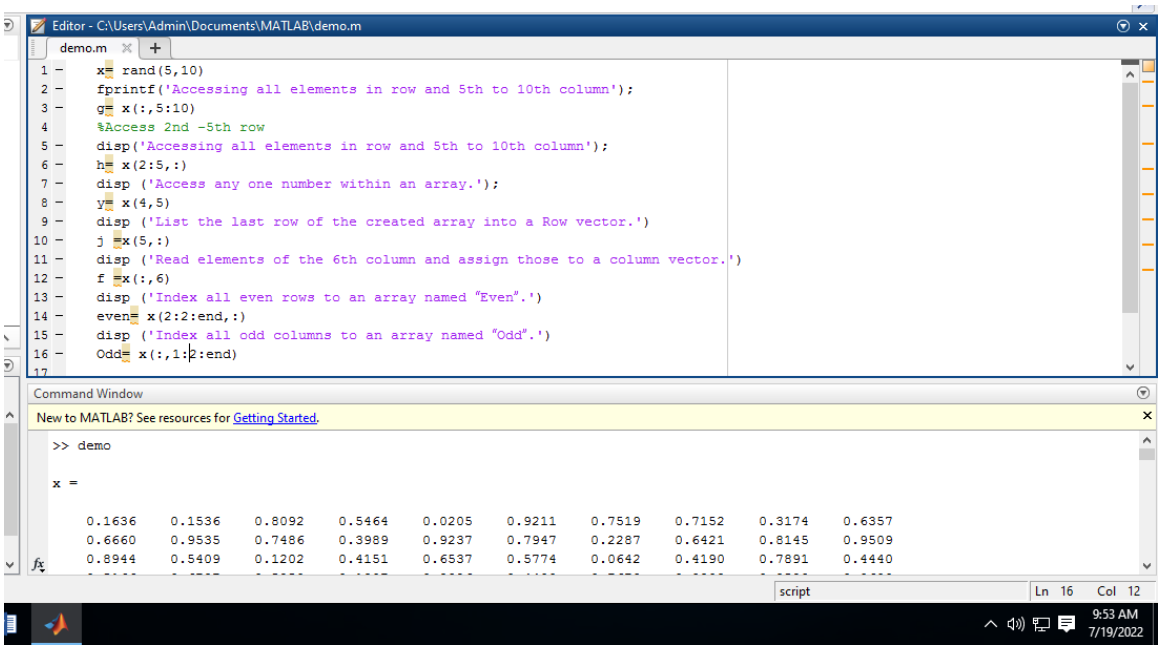
Step 9: Also, if want to choose the odd columns in the array, we have to just enter 1:2:end so this simply means that the range will start from index 1 with a gap of 2.

Step 10: Now your result would be printed in the command window and the comments can be added using % sign if you want to.

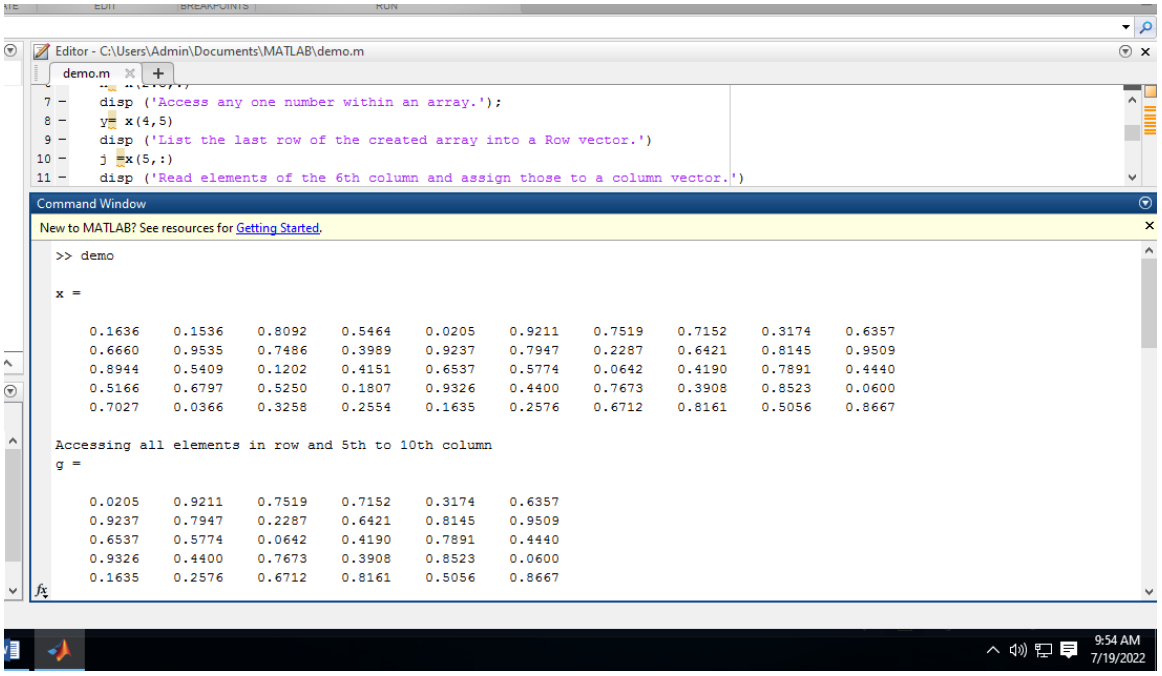
Step 11: Stop

1. **Access 5th -10th column.**

Code:

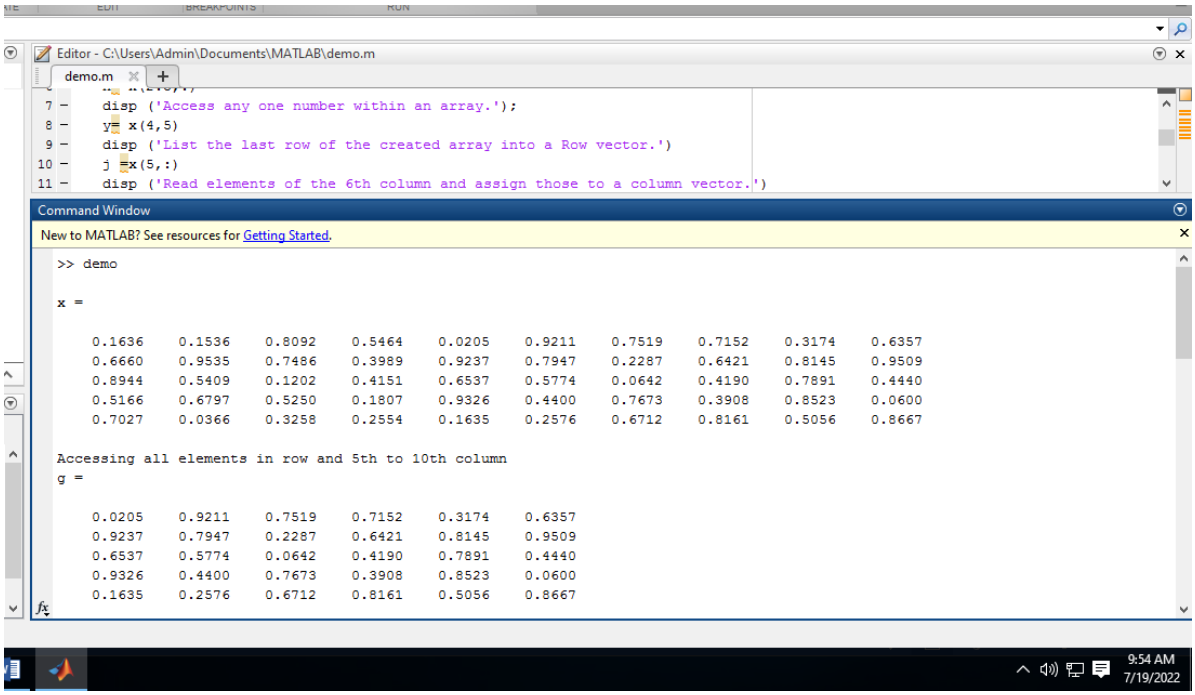


Output:

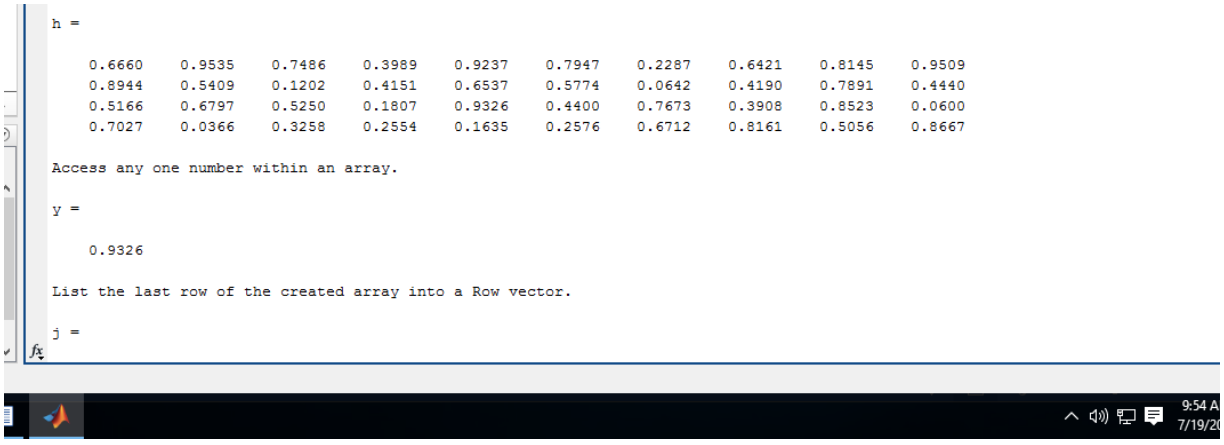


2. Access 2nd -5th row.

Code:

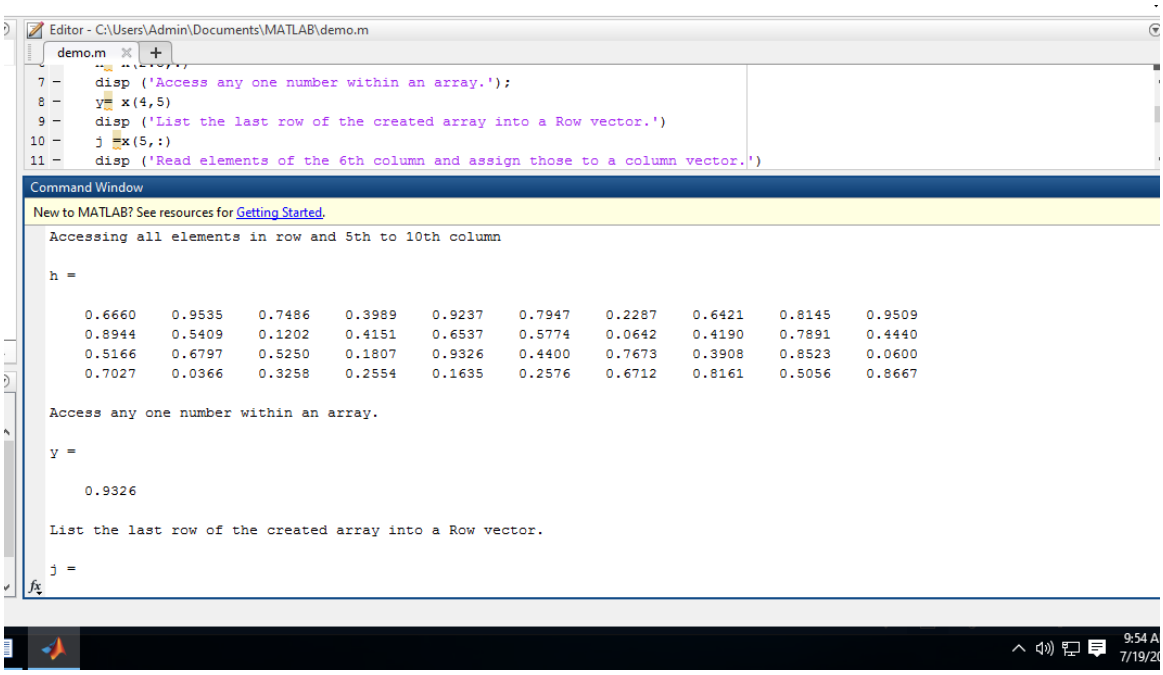


Output:

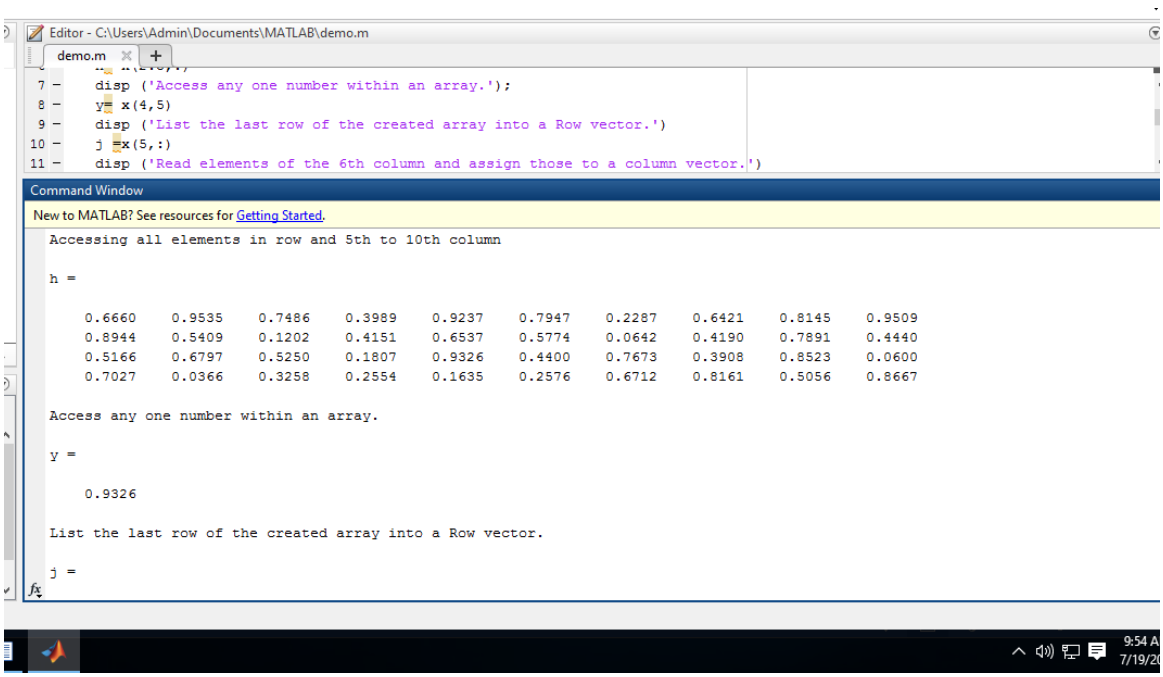


3. Access any one number within an array.

Code:

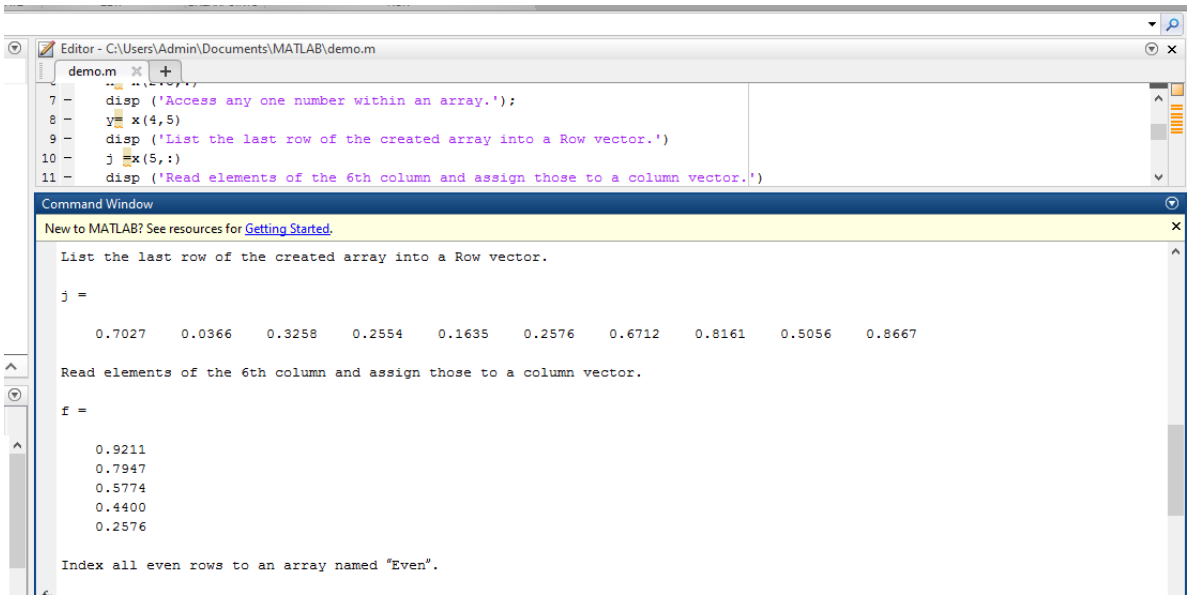


Output:

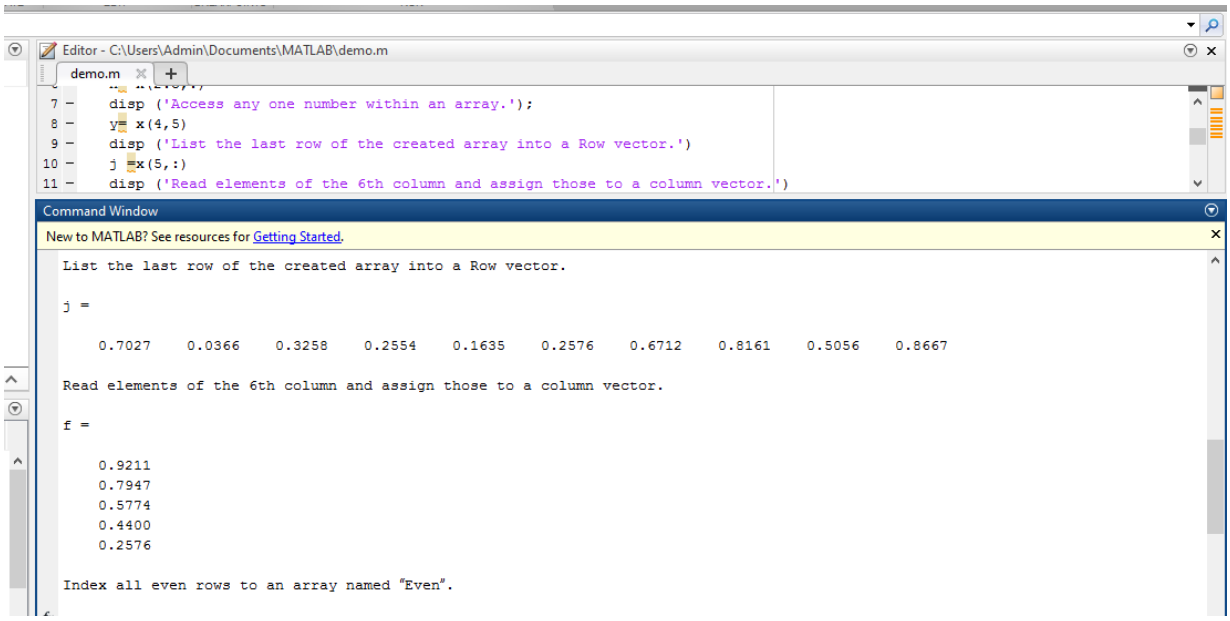


4. List the last row of the created array into a Row vector.

Code:

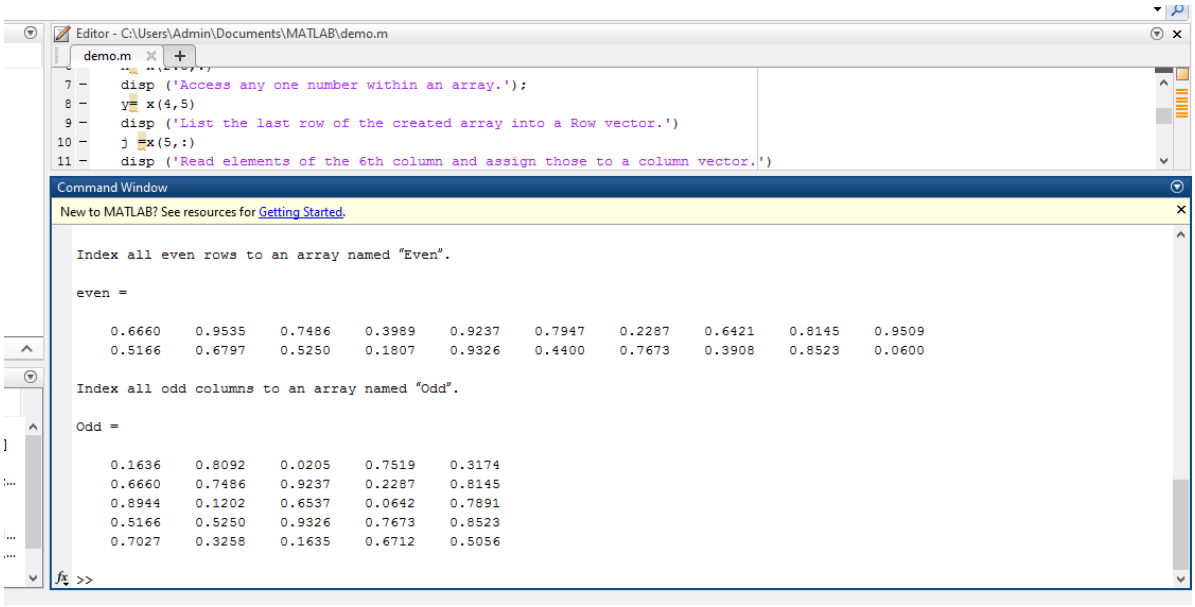


Output:

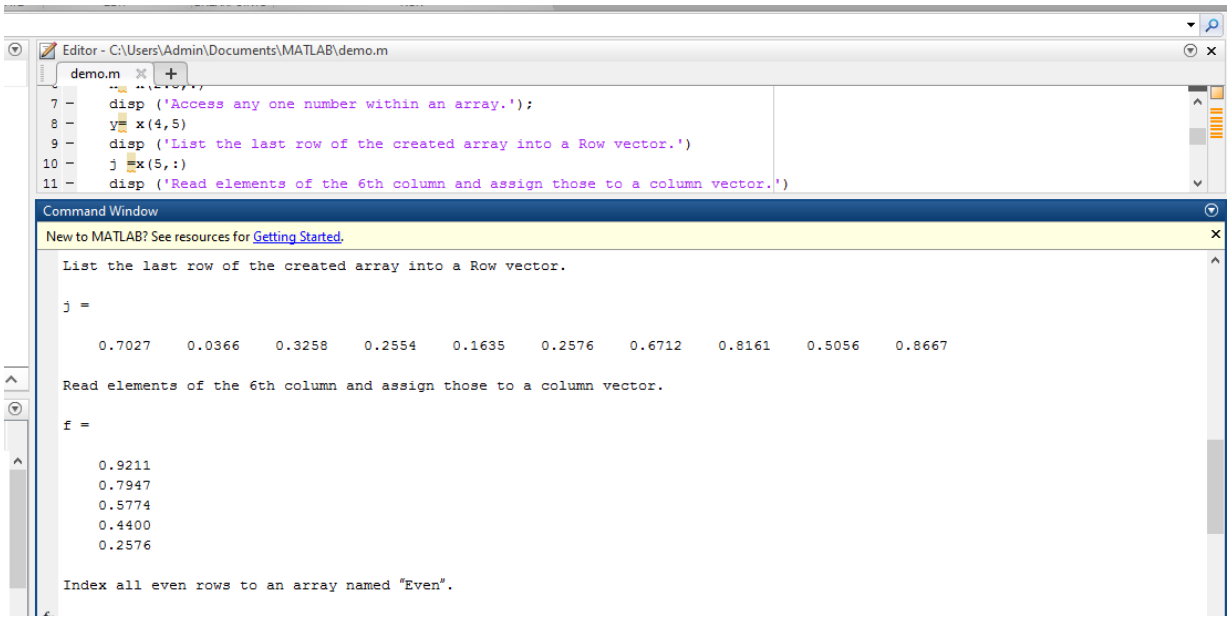


5. Read elements of the 6th column and assign those to a column vector.

Code:

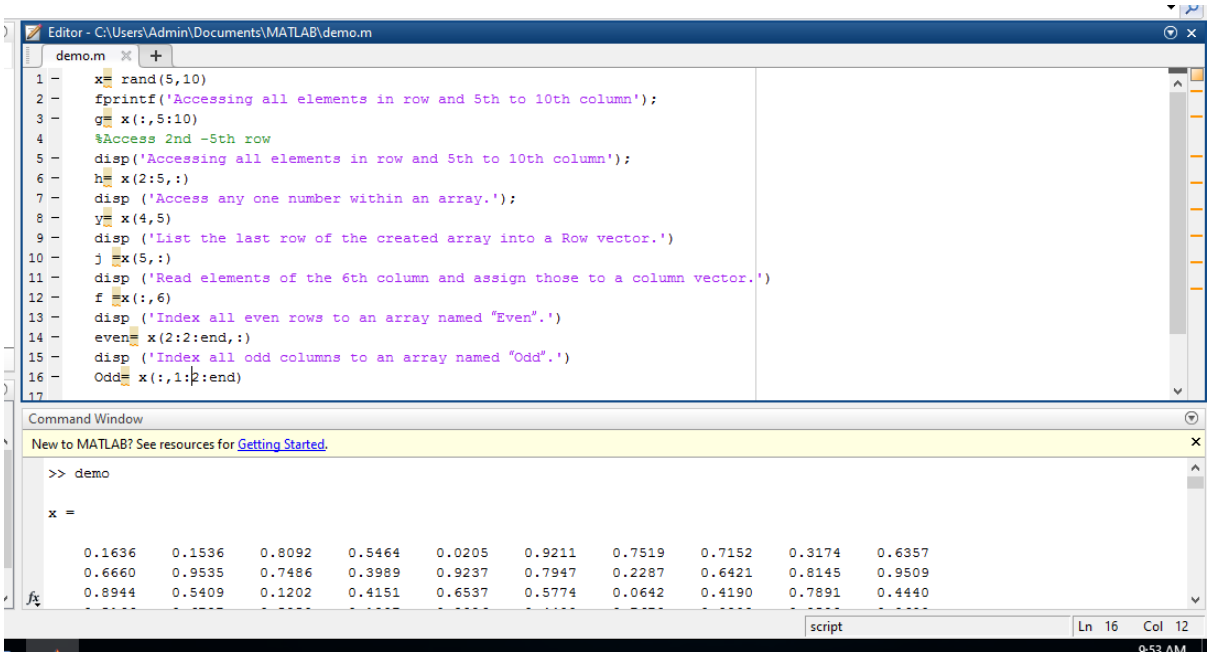


Output:

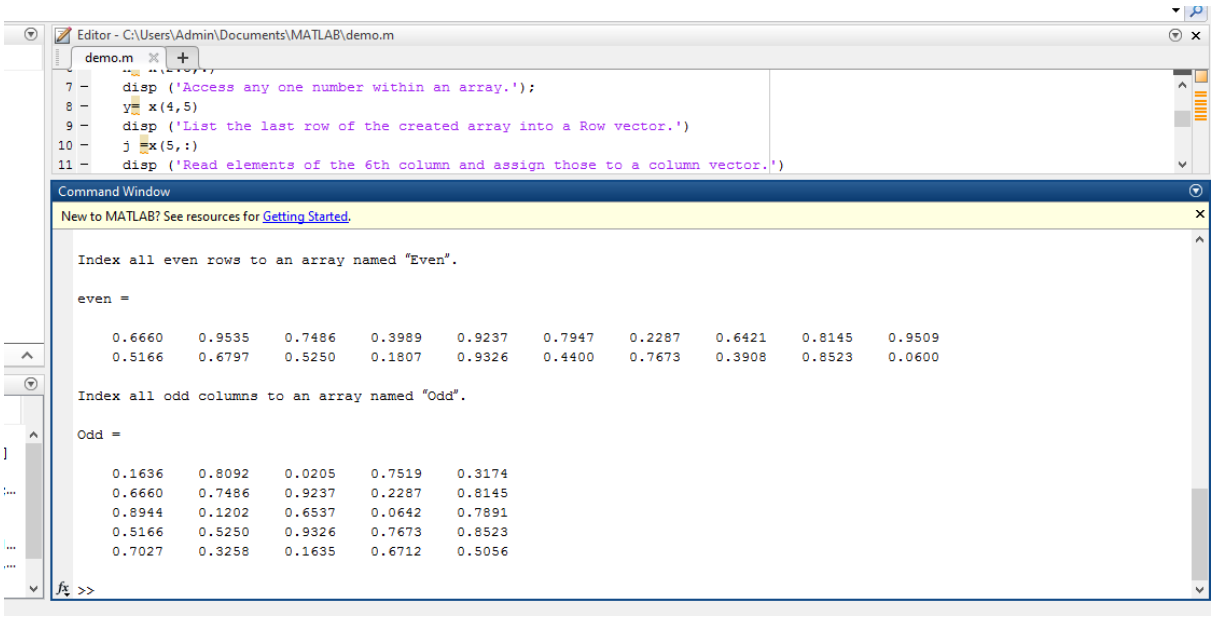


6. Index all even rows to an array named “Even”.

Code:

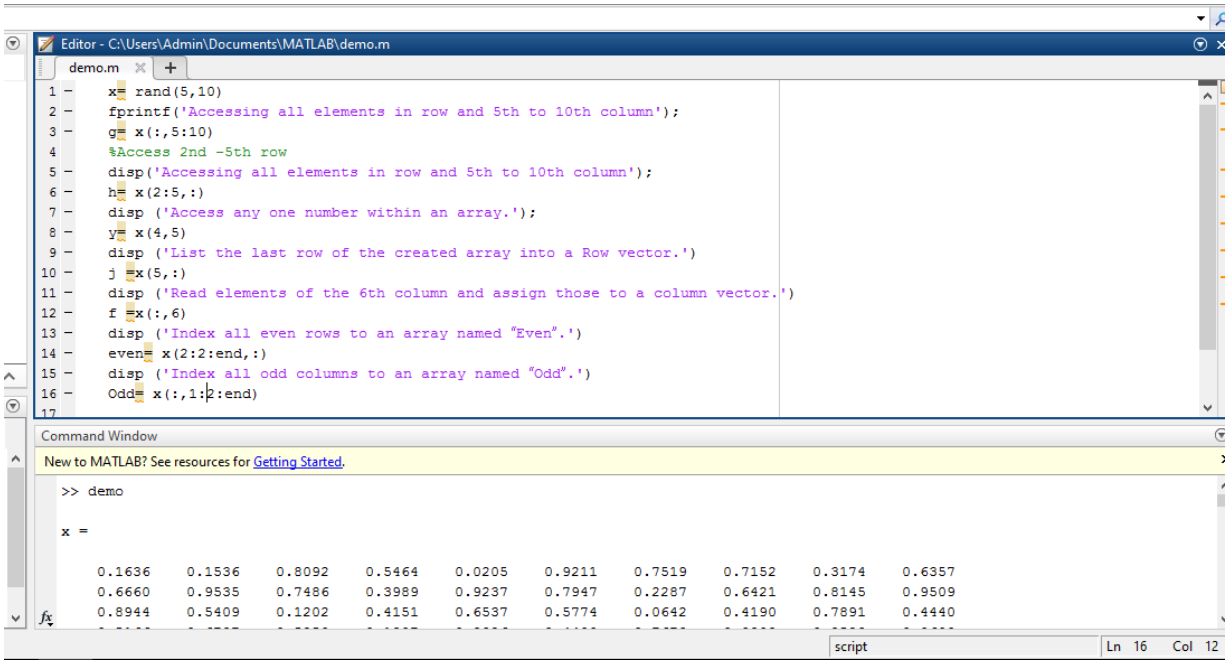


Output:

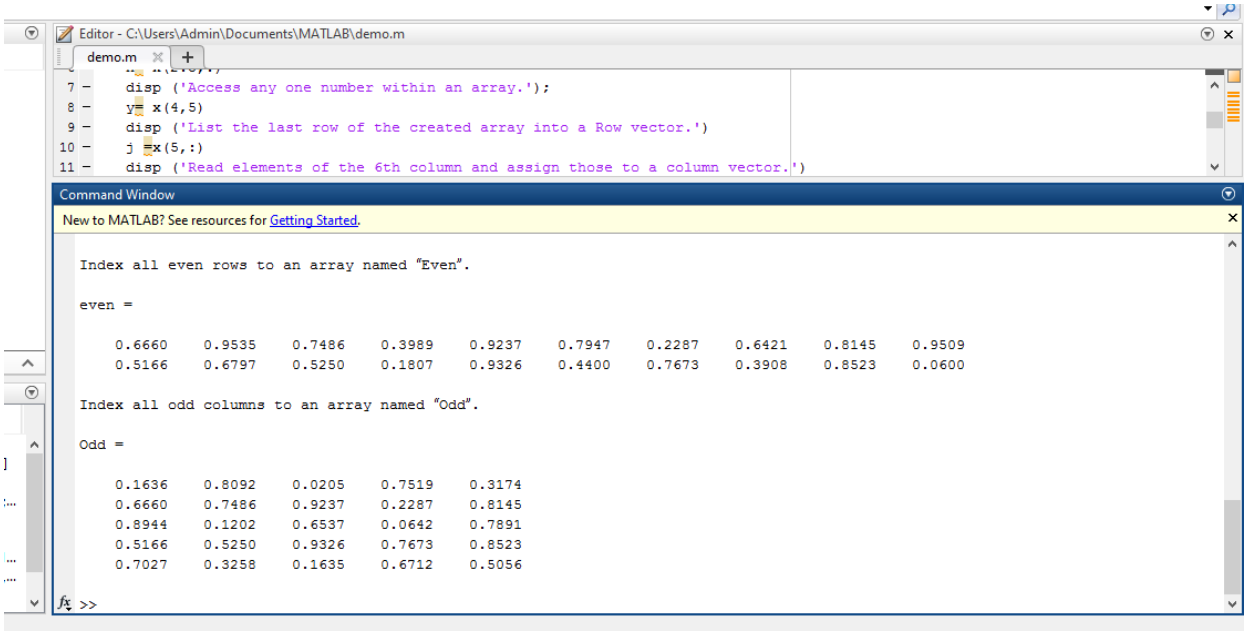


7. Index all odd columns to an array named “Odd”.

Code:



Output:



Q2. Write a program to read all the elements of the first seven columns from the previously created array (5X10 array created in the previous question), to perform the following operations: (Use new variables to store the values for each sub-question).

Logic:

Step 1: Start

Step 2: First we have to create a new script named as demo.m

Step 3: To find reciprocal, the function reci() is used.

Step 4: So, basically to find sum of 3rd column and 6th column we have to first access the columns and then find sum of the entire column using sum() and then add the sums of 2 columns.

Step 5: Now if you want to access any 4 columns and store it in a new variable we have to use reshape() to reshape the matrix.

Step 6: So, the Sqrt() is used to find square root of an array.

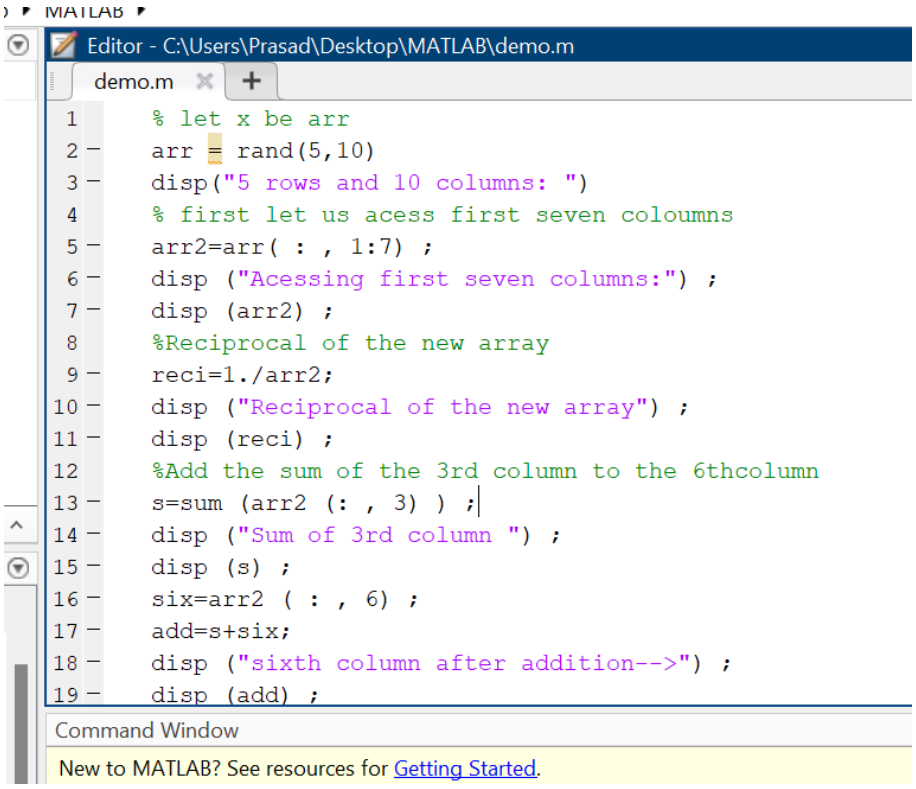
Step 7: We use ^ is used to find square of an array.

Step 8: Your result would be printed in the command window and the comments can be added using % sign if you wish to.

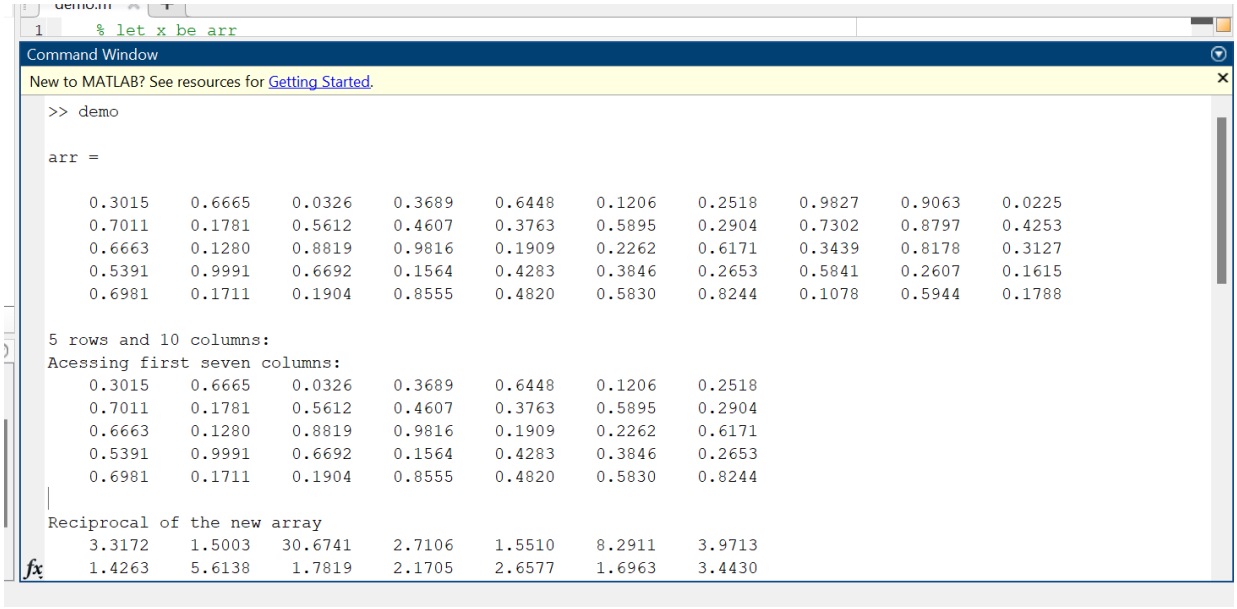
Step 9: Stop.

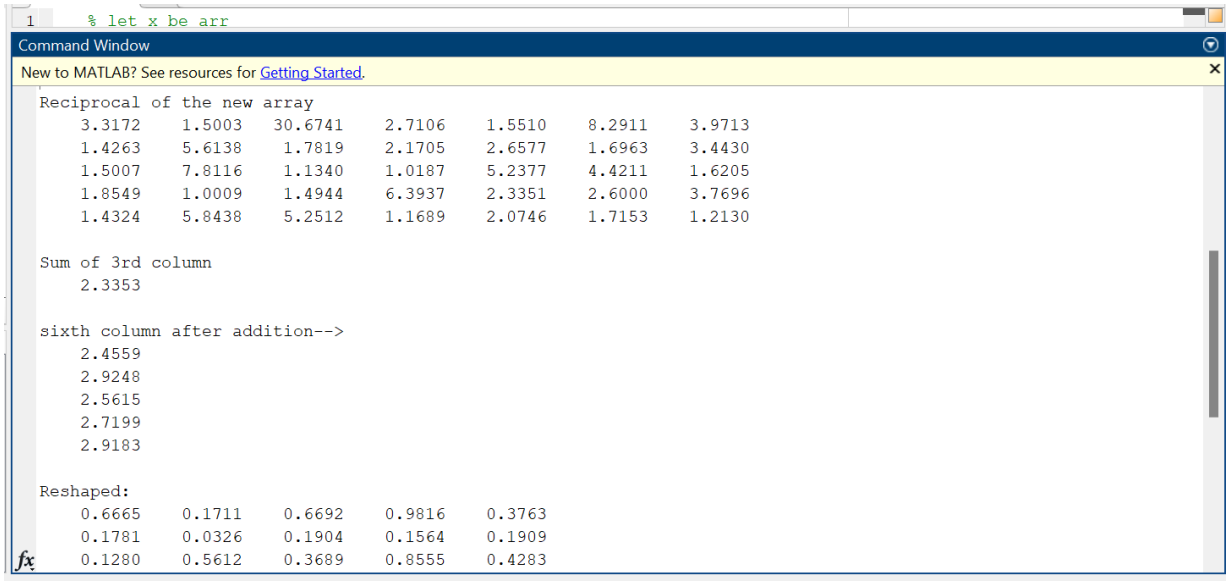
1. **Reciprocal of the new array.**

Code:



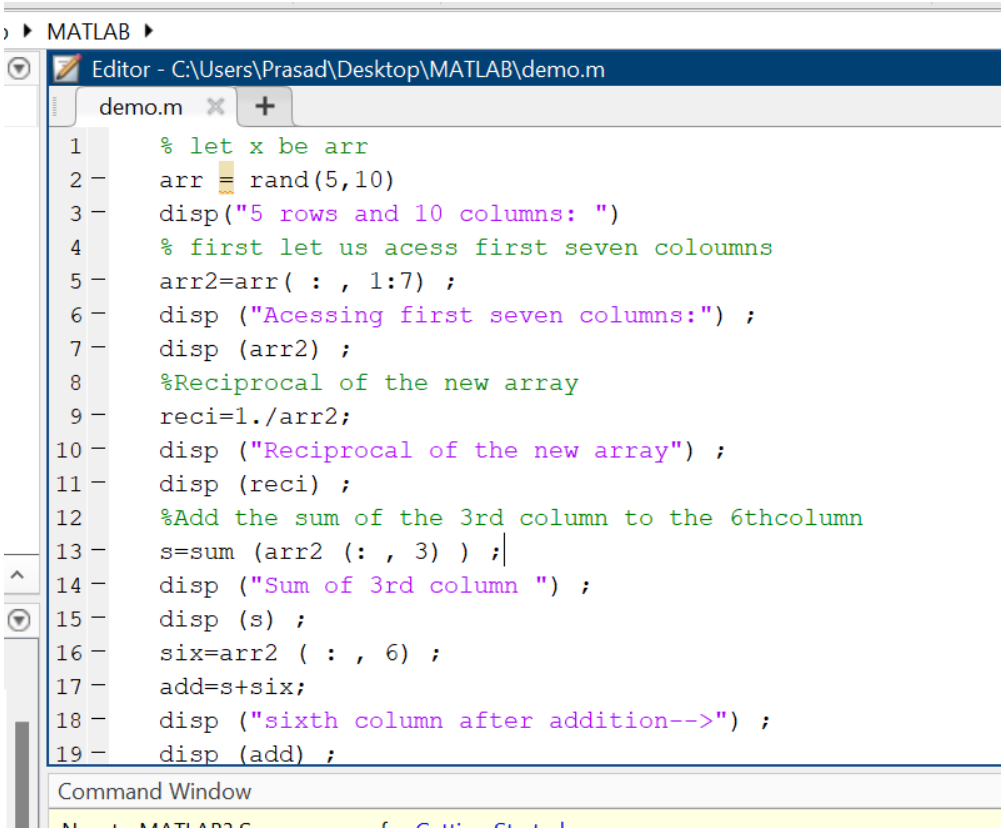
Output:


% let x be arr
Command Window
New to MATLAB? See resources for Getting Started.
>> demo
arr =
0.3015 0.6665 0.0326 0.3689 0.6448 0.1206 0.2518 0.9827 0.9063 0.0225
0.7011 0.1781 0.5612 0.4607 0.3763 0.5895 0.2904 0.7302 0.8797 0.4253
0.6663 0.1280 0.8819 0.9816 0.1909 0.2262 0.6171 0.3439 0.8178 0.3127
0.5391 0.9991 0.6692 0.1564 0.4283 0.3846 0.2653 0.5841 0.2607 0.1615
0.6981 0.1711 0.1904 0.8555 0.4820 0.5830 0.8244 0.1078 0.5944 0.1788
5 rows and 10 columns:
Acessing first seven columns:
0.3015 0.6665 0.0326 0.3689 0.6448 0.1206 0.2518
0.7011 0.1781 0.5612 0.4607 0.3763 0.5895 0.2904
0.6663 0.1280 0.8819 0.9816 0.1909 0.2262 0.6171
0.5391 0.9991 0.6692 0.1564 0.4283 0.3846 0.2653
0.6981 0.1711 0.1904 0.8555 0.4820 0.5830 0.8244
Reciprocal of the new array
3.3172 1.5003 30.6741 2.7106 1.5510 8.2911 3.9713
1.4263 5.6138 1.7819 2.1705 2.6577 1.6963 3.4430

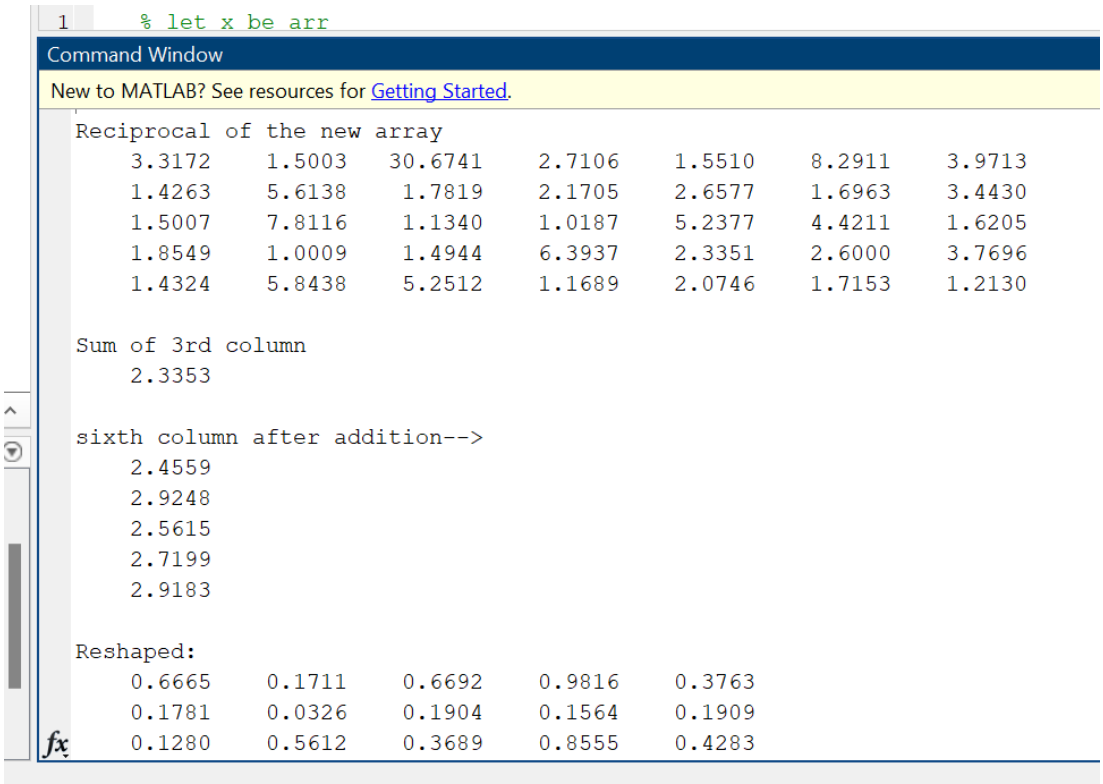

% let x be arr
Command Window
New to MATLAB? See resources for Getting Started.
Reciprocal of the new array
3.3172 1.5003 30.6741 2.7106 1.5510 8.2911 3.9713
1.4263 5.6138 1.7819 2.1705 2.6577 1.6963 3.4430
1.5007 7.8116 1.1340 1.0187 5.2377 4.4211 1.6205
1.8549 1.0009 1.4944 6.3937 2.3351 2.6000 3.7696
1.4324 5.8438 5.2512 1.1689 2.0746 1.7153 1.2130
Sum of 3rd column
2.3353
sixth column after addition-->
2.4559
2.9248
2.5615
2.7199
2.9183
Reshaped:
0.6665 0.1711 0.6692 0.9816 0.3763
0.1781 0.0326 0.1904 0.1564 0.1909
0.1280 0.5612 0.3689 0.8555 0.4283

2. Add the sum of the 3rd column to the 6th column.

Code:

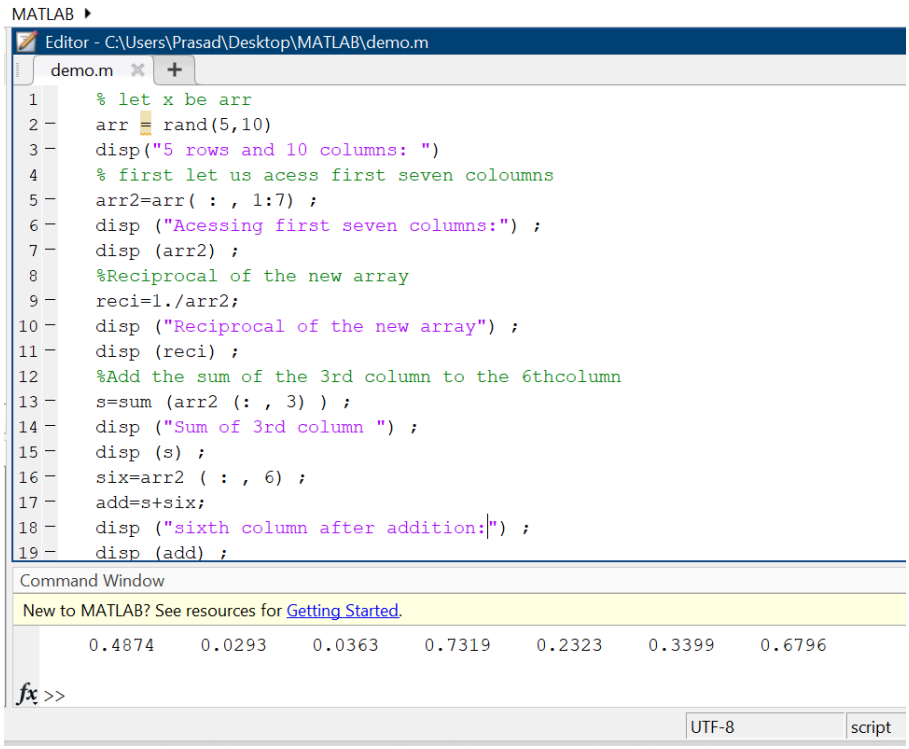


Output:

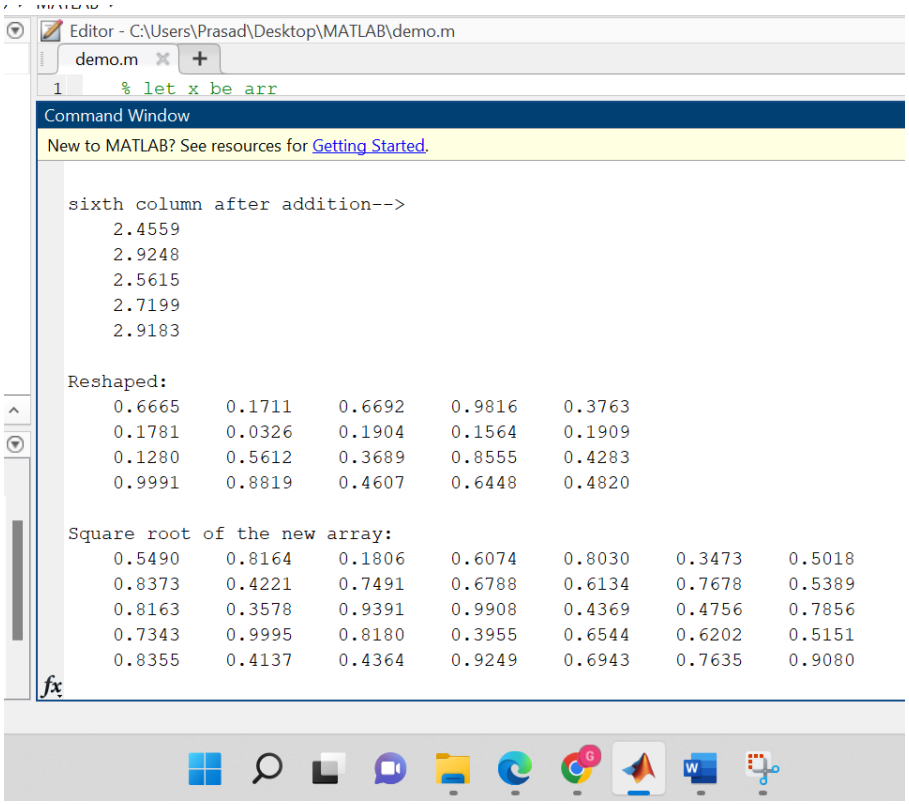


3. Access any of the four columns and reshape it to the 4X5 matrix.

Code:



Output:



4. Square root of the new array.

Code:

Editor - C:\Users\Prasad\Desktop\MATLAB\demo.m

demo.m

```
14 -    disp ("Sum of 3rd column ") ;
15 -    disp (s) ;
16 -    six=arr2 ( : , 6) ;
17 -    add=s+six;
18 -    disp ("sixth column after addition:") ;
19 -    disp (add) ;
20      %acess and reshape in 4x5
21 -    arr3=arr2 ( : , 2:5) ;
22 -    res=reshape (arr3, 4, 5) ;
23 -    disp ("Reshaped:") ;
24 -    disp (res) ;
25      %Square root of the new array
26 -    sq=sqrt (arr2) ;
27 -    disp ("Square root of the new array:") ;
28 -    disp (sq) ;
29      %Square of the new array
30 -    new= arr2.^2;
31 -    disp ("Square of the new array:") ;
32 -    disp (new) ;
```

Command Window

New to MATLAB? See resources for Getting Started.

```
    0.4874    0.0293    0.0363    0.7319    0.2323    0.3399    0.6796
```

Output:

EDIT BREAKPOINTS RUN

MATLAB

Editor - C:\Users\Prasad\Desktop\MATLAB\demo.m

demo.m

```
1       % let x be arr
```

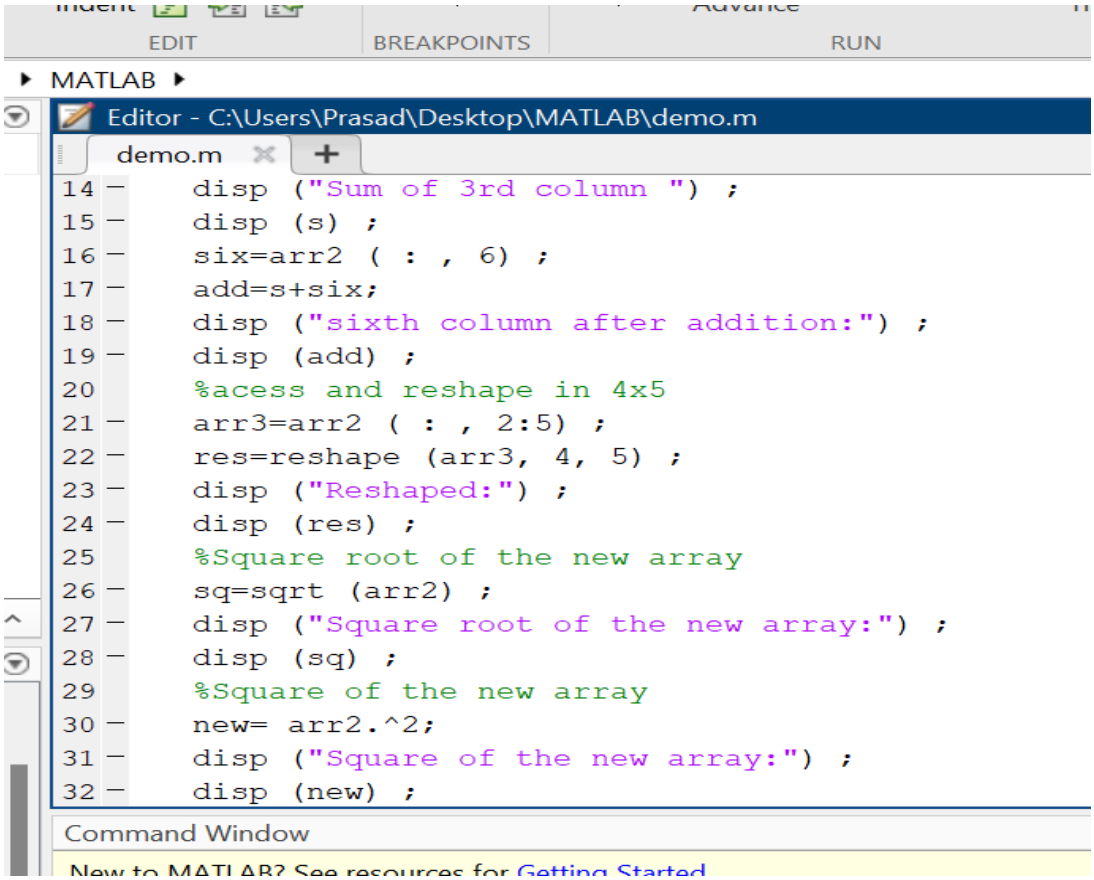
Command Window

New to MATLAB? See resources for Getting Started.

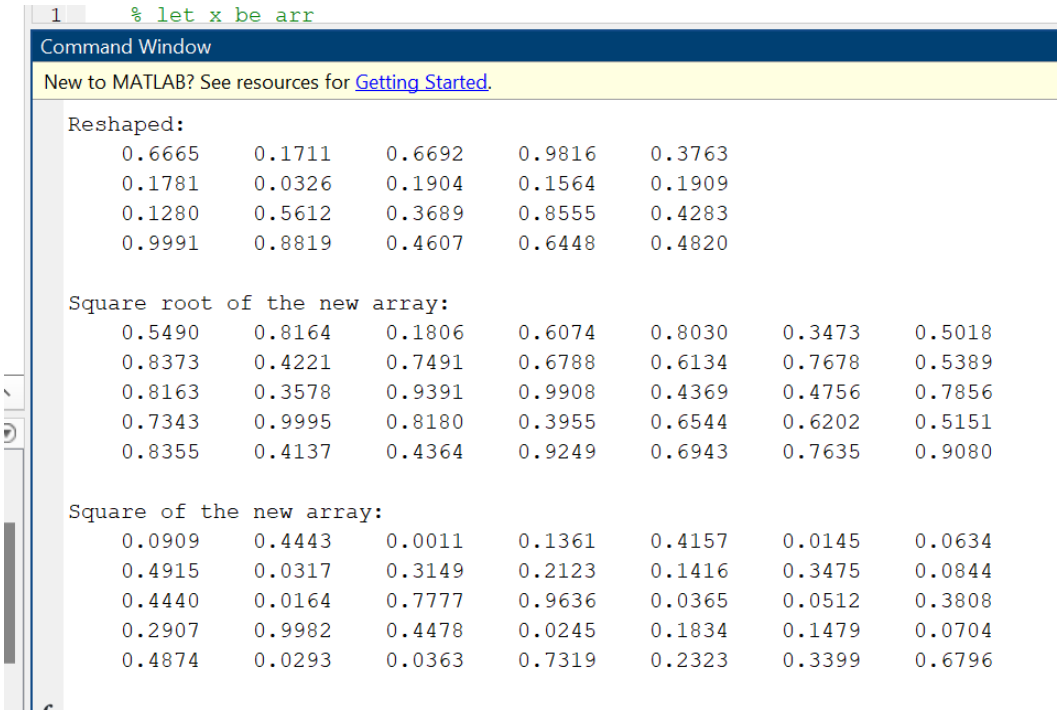
```
Reshaped:
    0.6665    0.1711    0.6692    0.9816    0.3763
    0.1781    0.0326    0.1904    0.1564    0.1909
    0.1280    0.5612    0.3689    0.8555    0.4283
    0.9991    0.8819    0.4607    0.6448    0.4820

Square root of the new array:
    0.5490    0.8164    0.1806    0.6074    0.8030    0.3473    0.5018
    0.8373    0.4221    0.7491    0.6788    0.6134    0.7678    0.5389
    0.8163    0.3578    0.9391    0.9908    0.4369    0.4756    0.7856
    0.7343    0.9995    0.8180    0.3955    0.6544    0.6202    0.5151
    0.8355    0.4137    0.4364    0.9249    0.6943    0.7635    0.9080

Square of the new array:
    0.0909    0.4443    0.0011    0.1361    0.4157    0.0145    0.0634
    0.4915    0.0317    0.3149    0.2123    0.1416    0.3475    0.0844
    0.4440    0.0164    0.7777    0.9636    0.0365    0.0512    0.3808
    0.2907    0.9982    0.4478    0.0245    0.1834    0.1479    0.0704
    0.4874    0.0293    0.0363    0.7319    0.2323    0.3399    0.6796

fx >>
```

5. Square of the new array.

Code:



Output:



Q3. Write a program to raise the power of the 1st column to the 4th column in a 3X6 array. (Use numbers 1-10)

Logic:

Step 1: Start

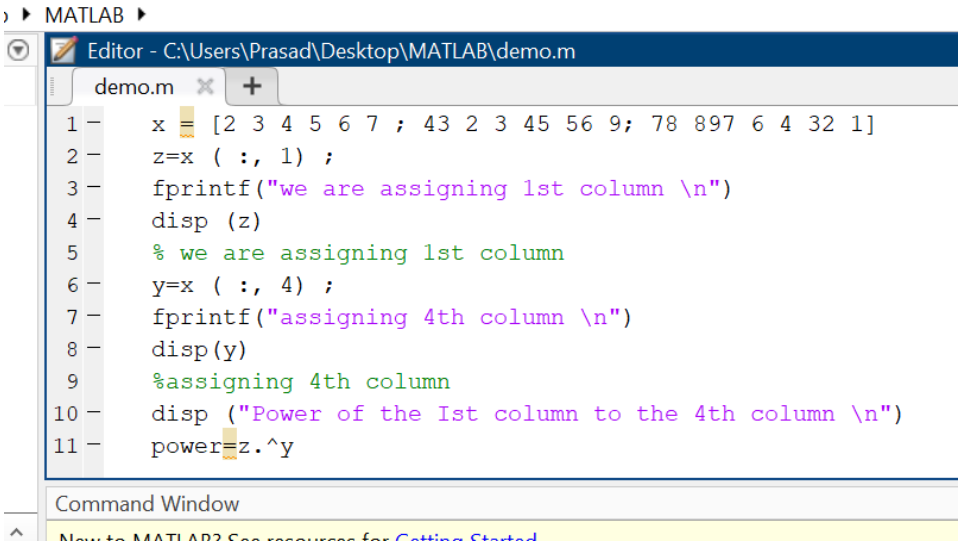
Step 2: First we have to create a new script named as demo.m.

Step 3: Now we have to create a 3X6 matrix. Access the 1st and 4th column.

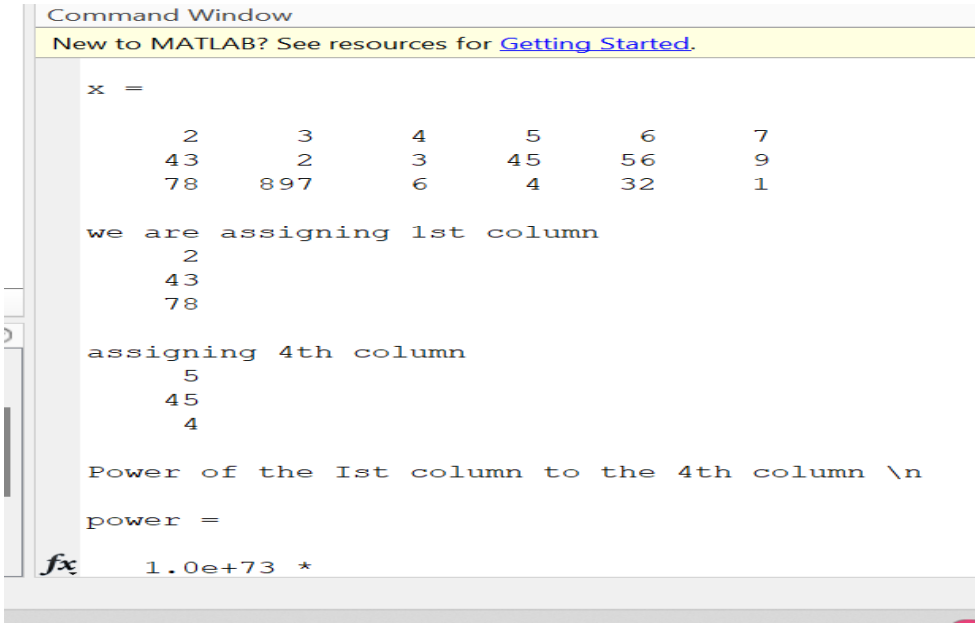
Step 4: So just use the . ^ is used to raise 1st to the 4th column.

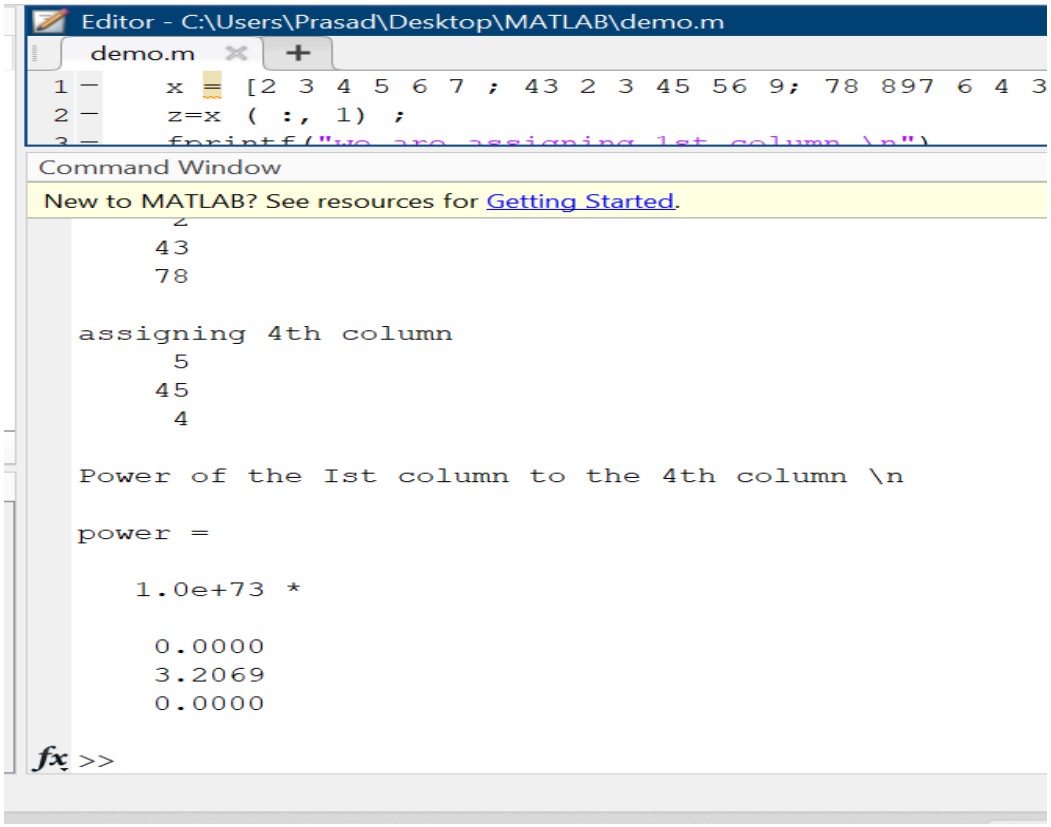
Step 5: Stop

Code:



Output:




Editor - C:\Users\Prasad\Desktop\MATLAB\demo.m
demo.m
1 x = [2 3 4 5 6 7 ; 43 2 3 45 56 9; 78 897 6 4 3
2 z=x (:, 1) ;
Command Window
New to MATLAB? See resources for Getting Started.
43
78
assigning 4th column
5
45
4
Power of the Ist column to the 4th column \n
power =
1.0e+73 *
0.0000
3.2069
0.0000
>>

Q4. Write a program to calculate the sum of each column of an array of size 5X5 and assign those values to the last row of the array and display the size of the array.

Logic:

Step 1: Start

Step 2: So first we have to create a new script named as demoo.m

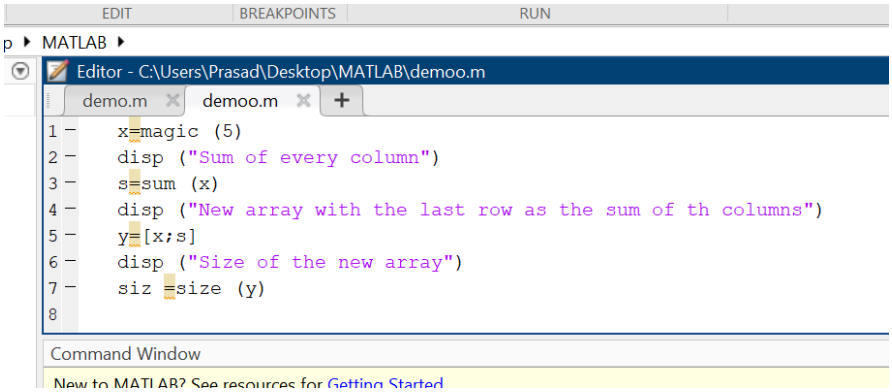
Step 3: Now we have to create a 5X5 matrix also we have to use sum() to calculate the sum of every column and assign it to a variable.

Step 4: Now we just have to create a new matrix and append the sum to it.

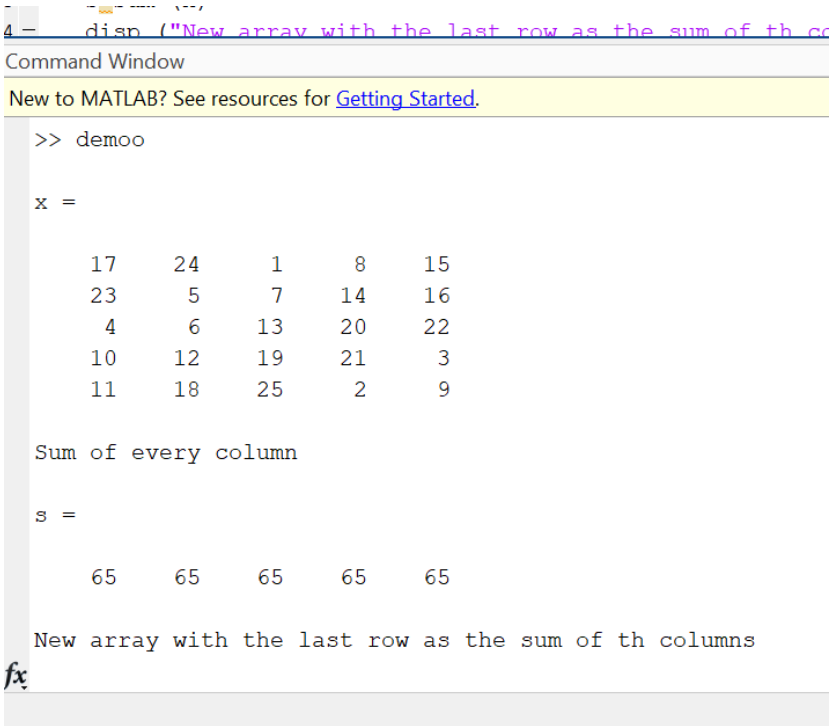
Step 5: So, the result would be printed in the command window.

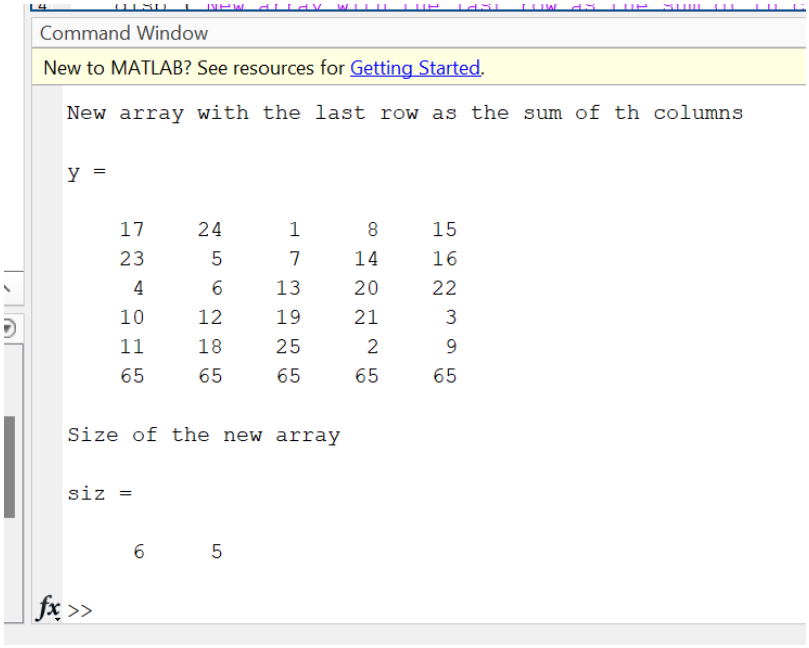
Step 6: Stop

Code:



Output:




Command Window
New to MATLAB? See resources for Getting Started.
New array with the last row as the sum of th columns
y =
17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9
65 65 65 65 65
Size of the new array
siz =
6 5
>>

Q5. Write a program to use the "Sort" function to sort the array of size 4X4. Sort row-wise and column-wise.

Logic:

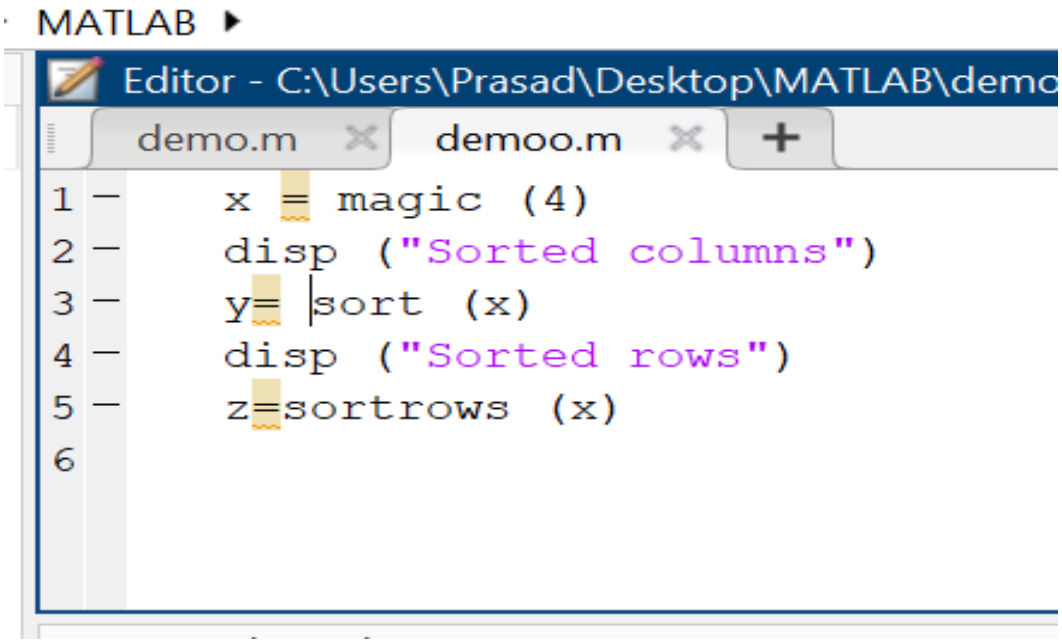
Step 1: Start

Step 2: First we have to Create a new script named as demoo.m

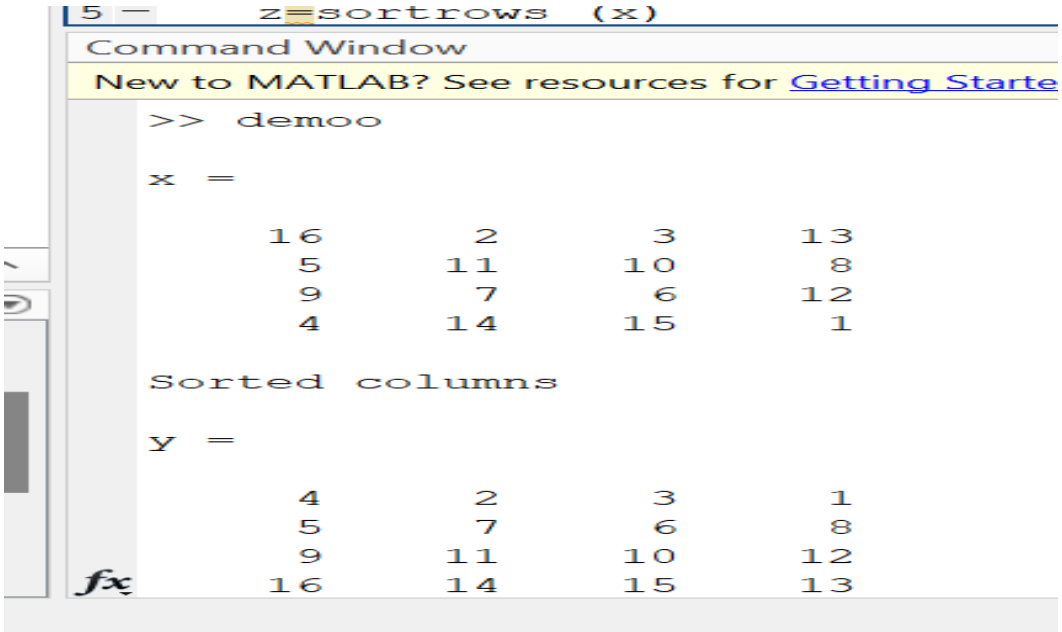
Step 3: Now, just create a 5X5 matrix and we have to use sort() that sorts the columns and sortrows() sorts the rows.

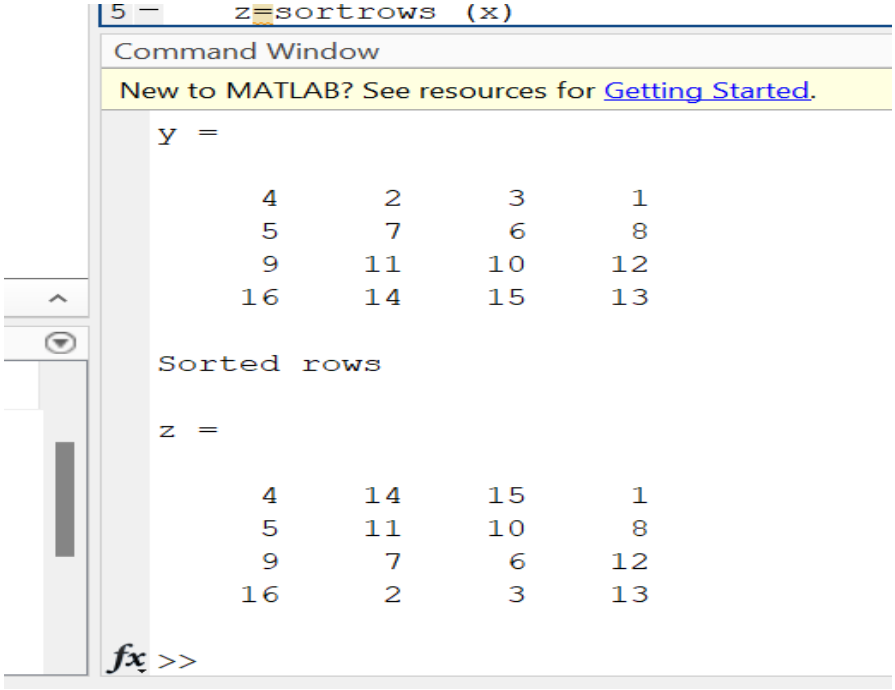
Step 4: Stop

Code:



Output:




5 — z=sortrows (x)
Command Window
New to MATLAB? See resources for Getting Started.
y =
4 2 3 1
5 7 6 8
9 11 10 12
16 14 15 13
Sorted rows
z =
4 14 15 1
5 11 10 8
9 7 6 12
16 2 3 13
>>

Q6. Write a program to use the "Circshift" function to move the array elements in a circular pattern and bring it back to its normal position.

Logic:

Step 1: Start

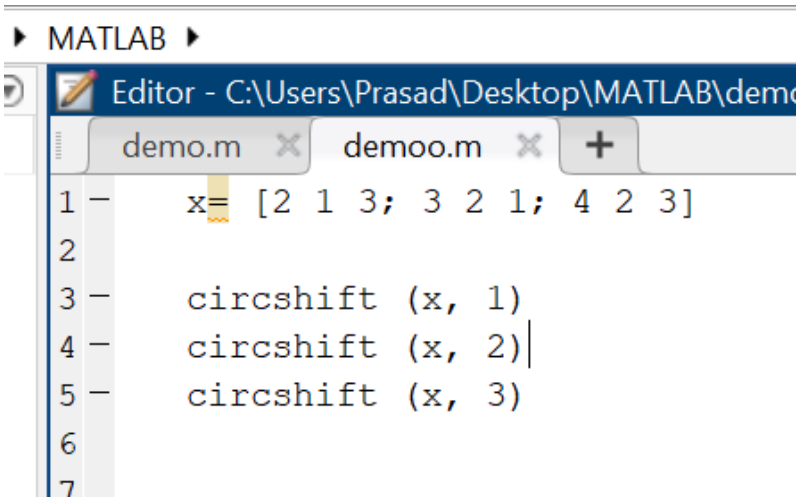
Step 2: So first we have to create a new script named as demoo.m

Step 3: So, now we have to create a matrix and use Circhsift() that shifts the rows.

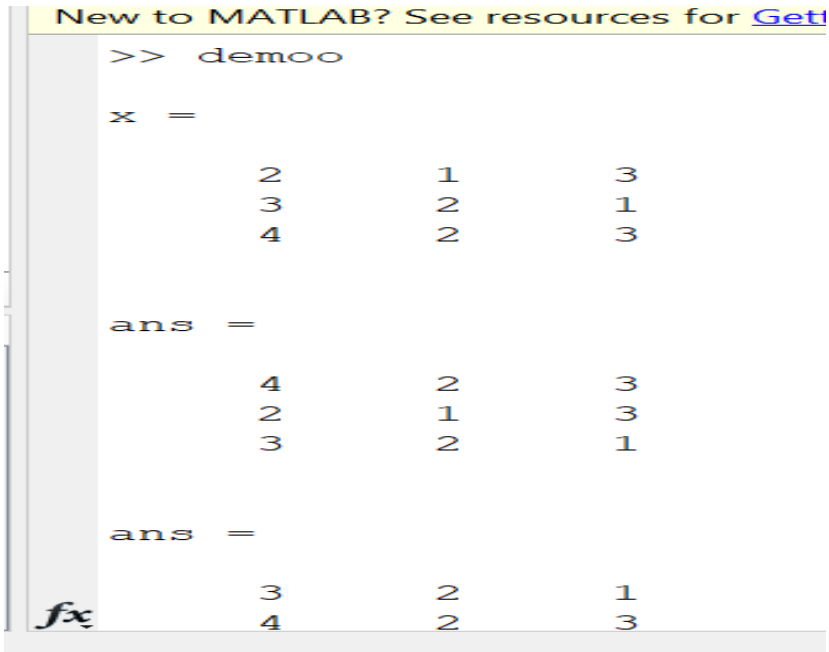
Step 4: So, we have to now see that it moves the last row to the top and moves the other rows down by 1 index.

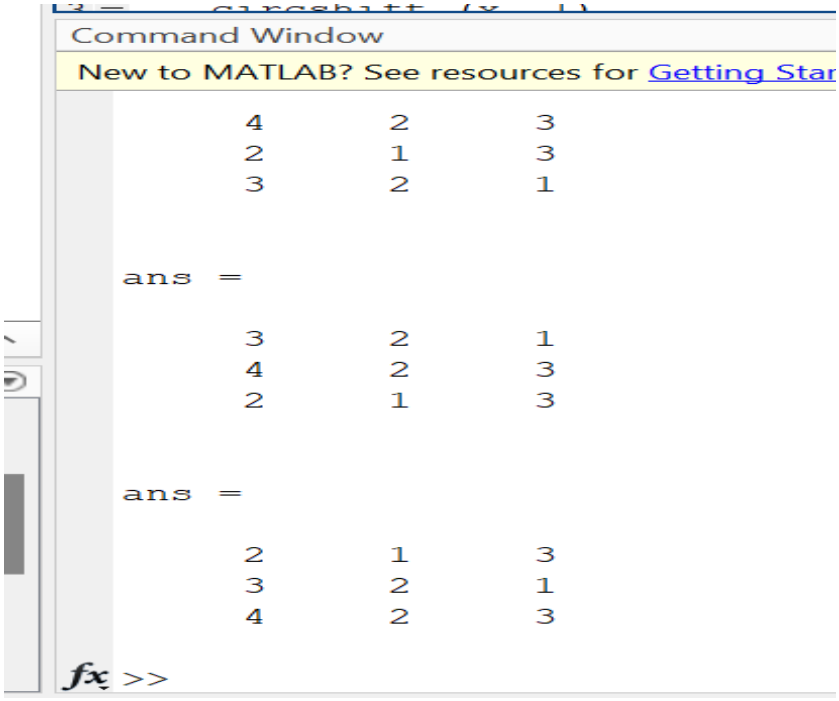
Step 5: Stop

Code:



Output:




Command Window
New to MATLAB? See resources for Getting Star
4 2 3
2 1 3
3 2 1
ans =
3 2 1
4 2 3
2 1 3
ans =
2 1 3
3 2 1
4 2 3
>>

Q7. Write a program to display a maximum and minimum number within the created array of any size.

Logic:

Step 1: Start

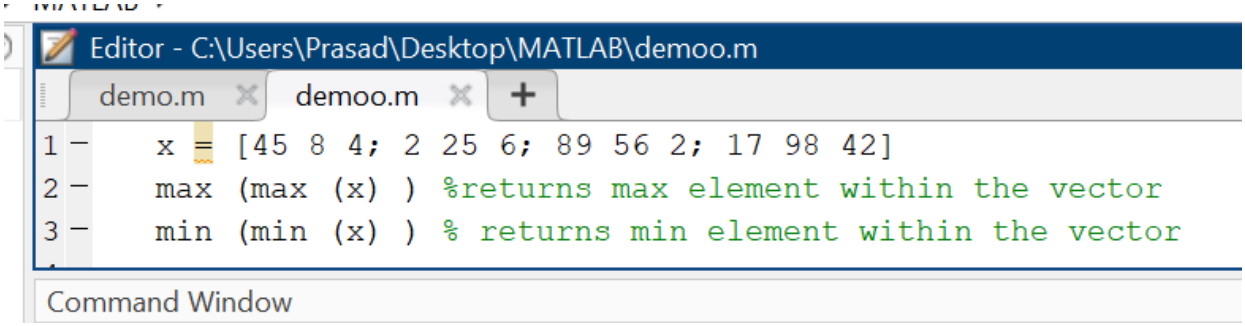
Step 2: So, first we have to create a new script named as demoo.m.

Step 3: So, the Max() and min() will give the entire row with maximum and minimum elements.

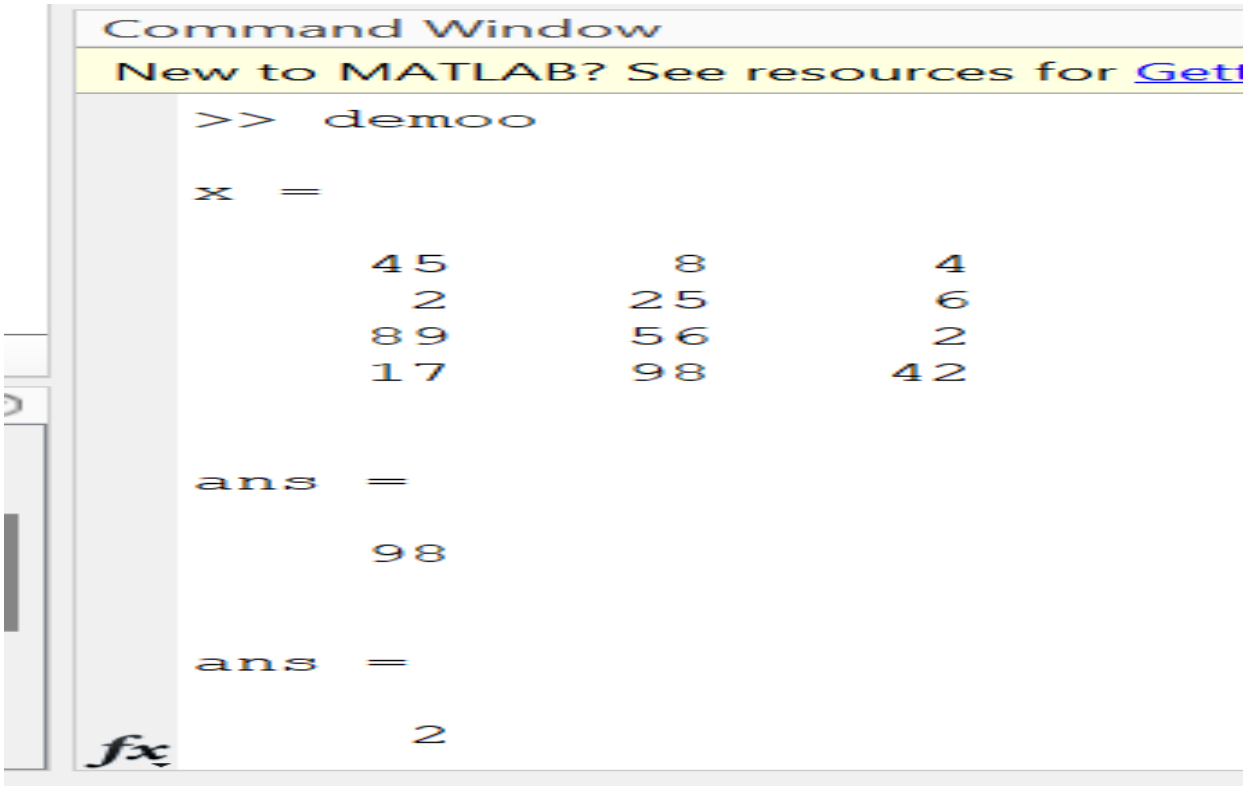
Step 4: Also, the Max(max()) and min(min()) will give the maximum and minimum element within the vector.

Step 5: Stop

Code:



Output:



Q8. Write a program to create a random array of 10 rows and 5 columns to display elements less than 0.89.

Logic:

Step 1: Start

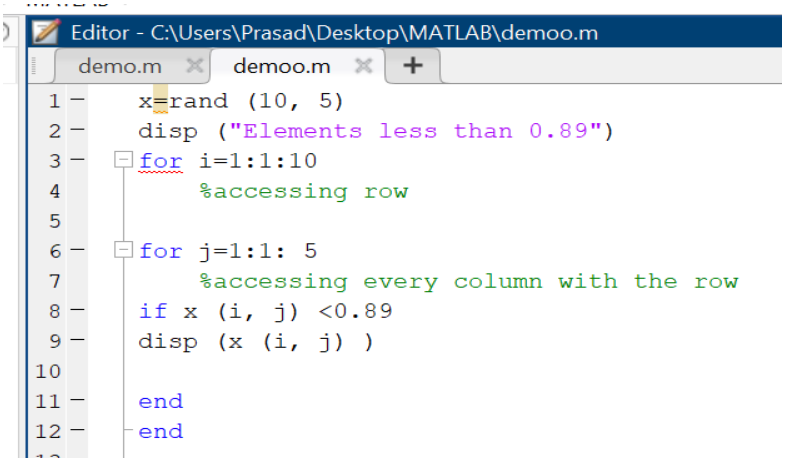
Step 2: First we have to create a new script named as demoo.m

Step 3: Now we have to first create a matrix with rand() to create a matrix with random functions.

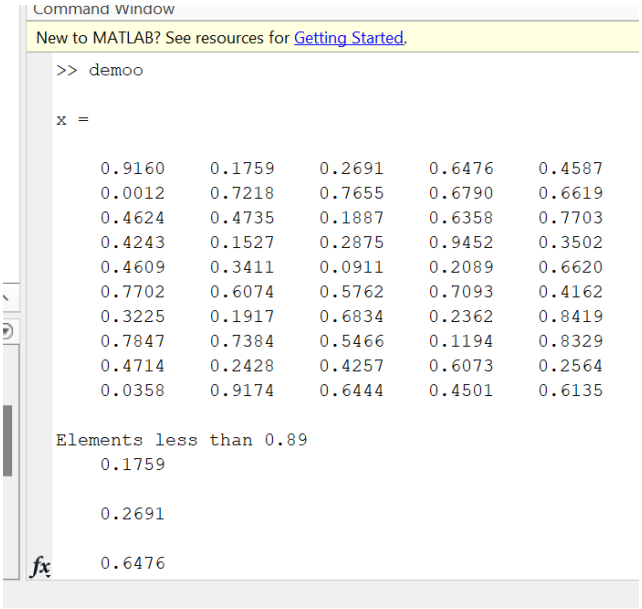
Step 4: After that we have to find elements less than 0.89, access the row and then each column with the row using for loop so, then use conditional operator to verify if the element is less than 0.89.

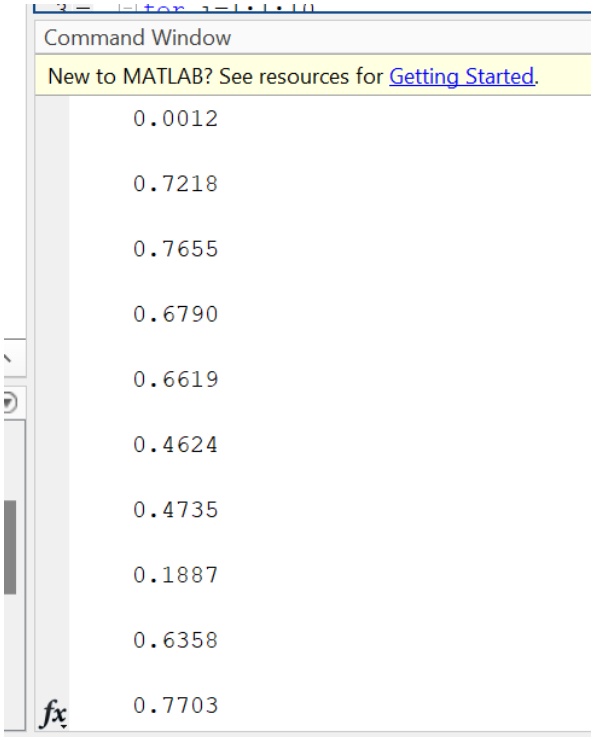
Step 5: Stop

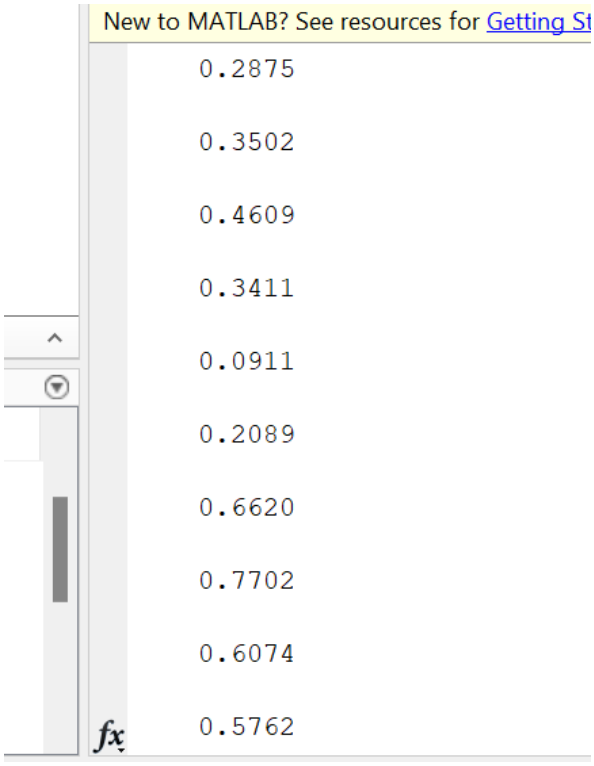
Code:



Output:




Command Window
New to MATLAB? See resources for Getting Started.
0.0012
0.7218
0.7655
0.6790
0.6619
0.4624
0.4735
0.1887
0.6358
0.7703


New to MATLAB? See resources for Getting St
0.2875
0.3502
0.4609
0.3411
0.0911
0.2089
0.6620
0.7702
0.6074
0.5762

Command Window

New to MATLAB? See resources for Getting Started.

```
    0.4714

    0.2428

    0.4257

    0.6073

    0.2564

    0.0358

    0.6444

    0.4501

    0.6135

fx >>
```

Q9. Write a program to create a 2D array with elements (7 10 5; 4 1 2; 3 6 9) to perform the following operations:

Logic:

Step 1: Start

Step 2: First we have to create a new script named demoo.m

Step 3: So, the (end, end, 2)=0 would create a new array in the 2nd dimension.

Step 4: After that the Cat(3,x,[…]) will create a new array in the 3rd dimension.

Step 5: Now the x(1,2,3) will give the element in the 1st row and 2nd column of the 3rd dimension.

Step 6: So, after that the x(2,3,3) will give the element in the 2nd row and 3rd column of the 3rd dimension.

Step 7: Here, the x(3,:,2) will give the element in the 3rd row of the 2nd dimension.

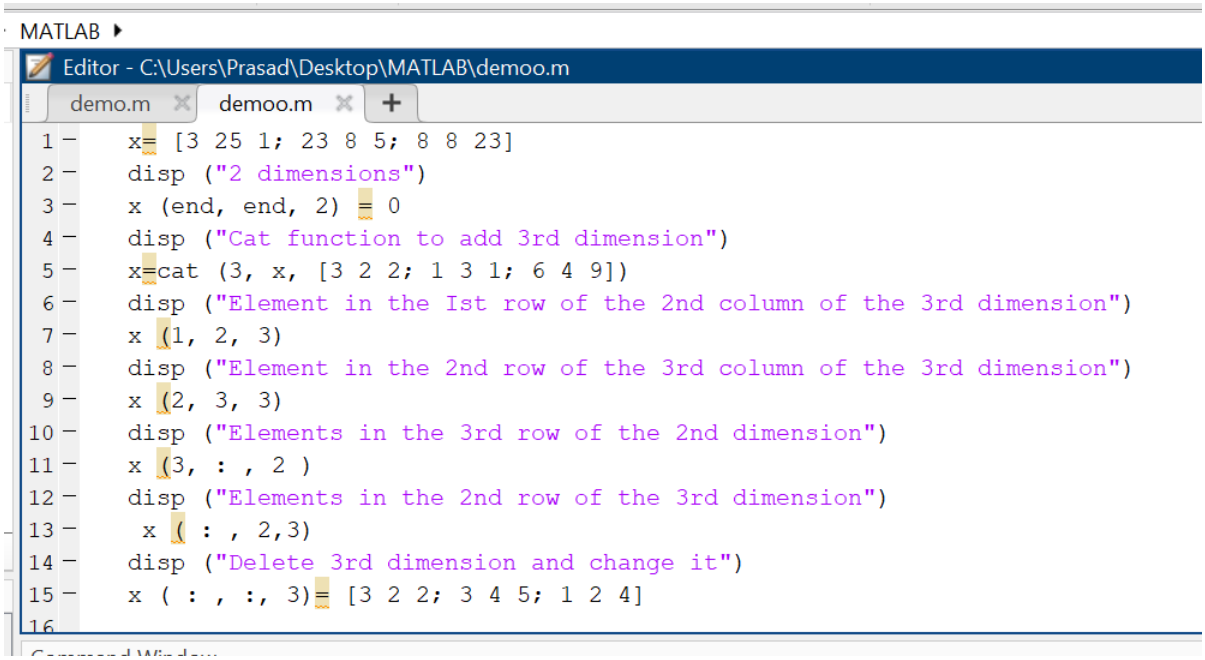
Step 8: After step 7 the x(:,2,3) will give the element in the 2nd column of the 3rd dimension.

Step 9: We have to now assign a new array to the 3rd dimension.

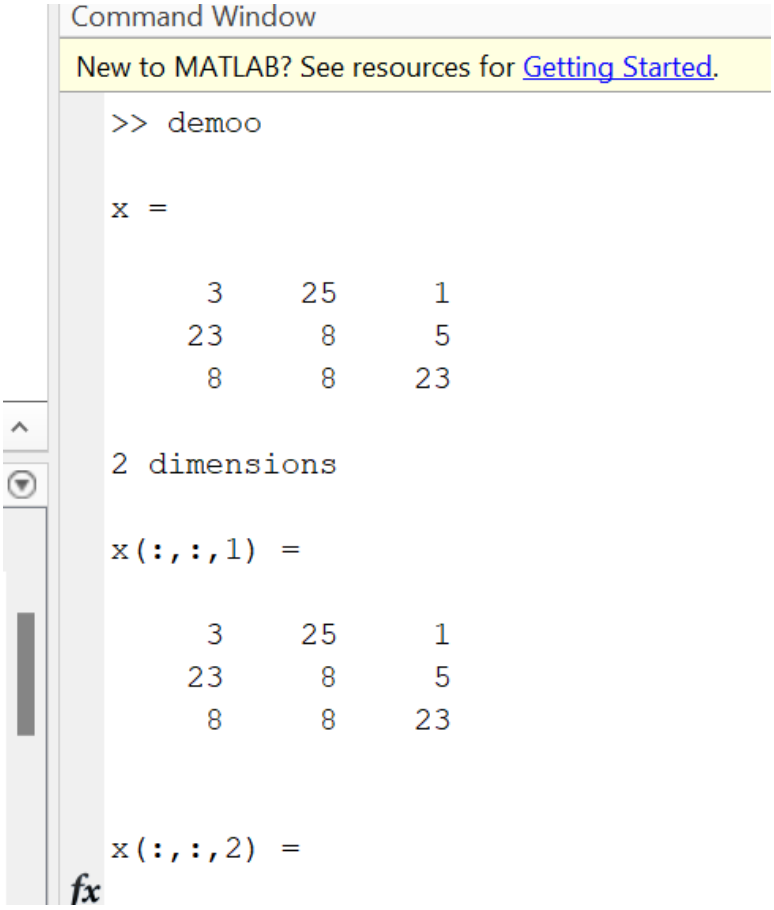
Step 10: Stop

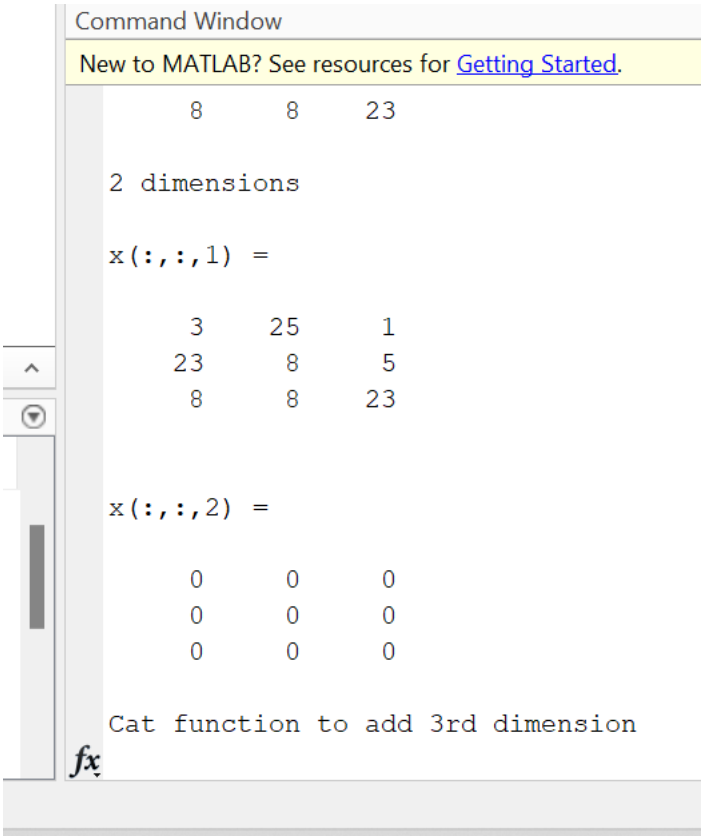
1. Add the 2nd dimension.

Code:



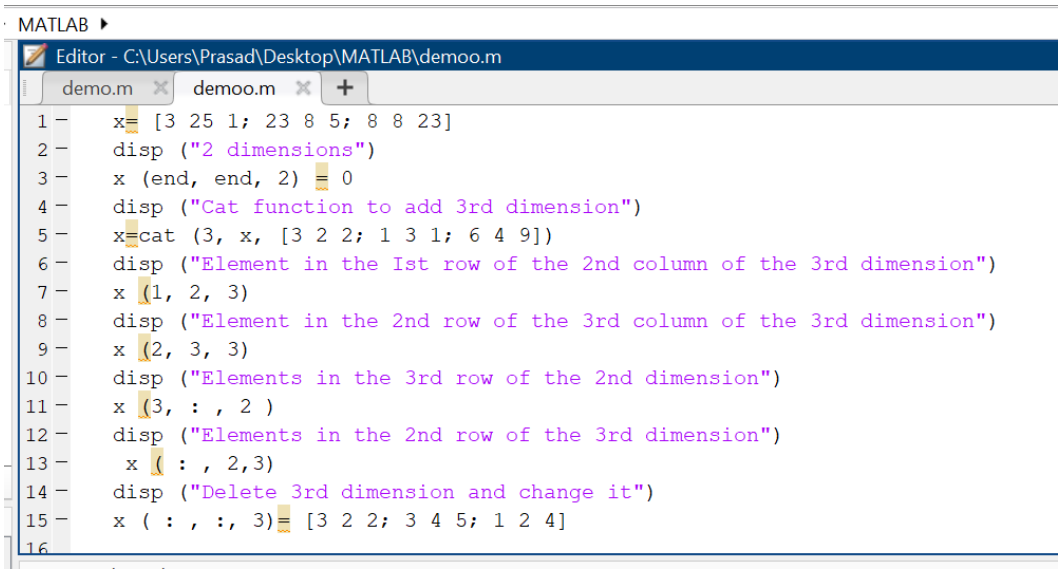
Output:


Command Window
New to MATLAB? See resources for Getting Started.
>> demoo
x =
3 25 1
23 8 5
8 8 23
2 dimensions
x(:,:,1) =
3 25 1
23 8 5
8 8 23
x(:,:,2) =

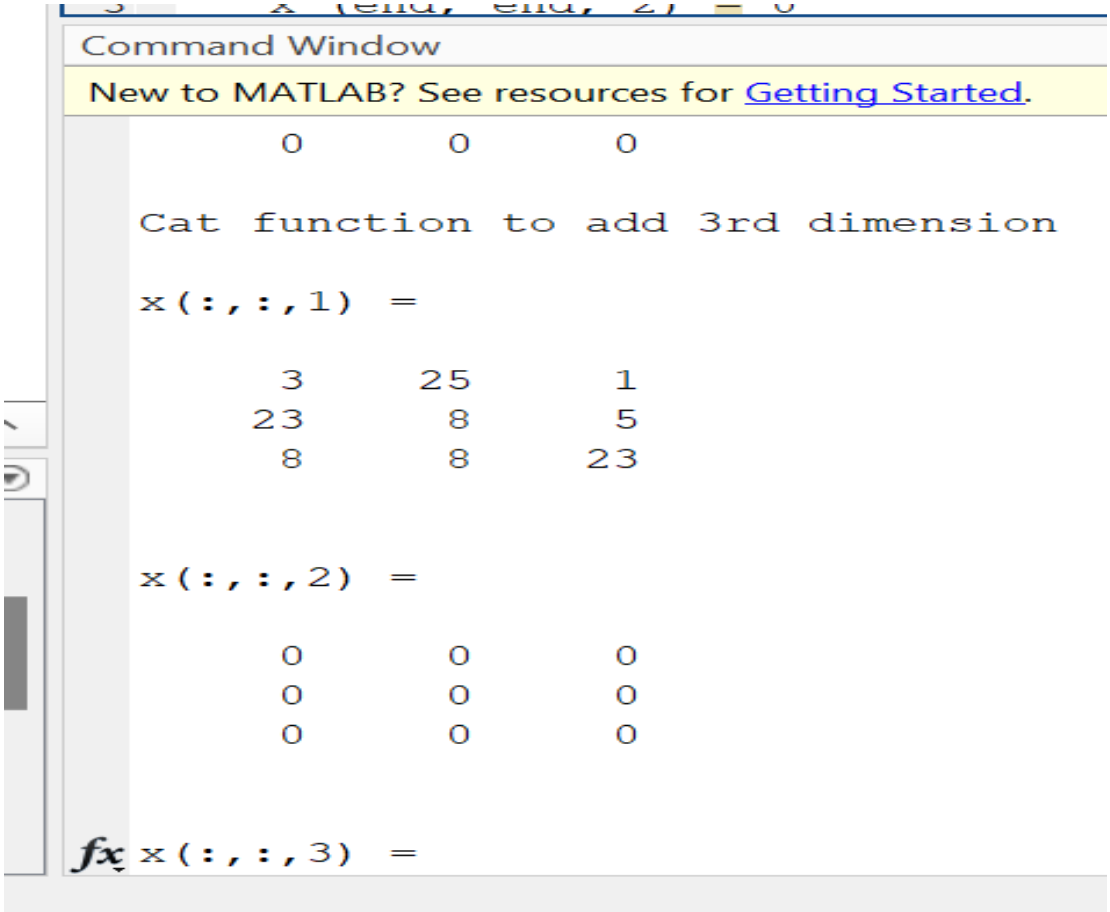

Command Window
New to MATLAB? See resources for Getting Started.
8 8 23
2 dimensions
x(:,:,1) =
3 25 1
23 8 5
8 8 23
x(:,:,2) =
0 0 0
0 0 0
0 0 0
Cat function to add 3rd dimension

2. Use the "Cat" function to add the 3rd dimension.

Code:



Output:



Editor - C:\Users\Prasad\Desktop\MATLAB\demoo.m

demo.m | demoo.m

```
1 -   x= [3 25 1; 23 8 5; 8 8 23]
2 -   disp ("2 dimensions")
3 -   x (end, end, 2) = 0
```

Command Window

New to MATLAB? See resources for Getting Started.

```
x(:,:,3) =

     3     2     2
     1     3     1
     6     4     9

Element in the Ist row of the 2nd column of the 3rd dimension

ans =

     2

Element in the 2nd row of the 3rd column of the 3rd dimension

ans =

     1

Elements in the 3rd row of the 2nd dimension
```

UTF-8

3. Access the element in the 1st row of the 2nd column of the 3rd dimension.

Code:

MATLAB

Editor - C:\Users\Prasad\Desktop\MATLAB\demoo.m

demo.m | demoo.m

```
x= [3 25 1; 23 8 5; 8 8 23]
disp ("2 dimensions")
x (end, end, 2) = 0
disp ("Cat function to add 3rd dimension")
x=cat (3, x, [3 2 2; 1 3 1; 6 4 9])
disp ("Element in the Ist row of the 2nd column of the 3rd dimension")
x (1, 2, 3)
disp ("Element in the 2nd row of the 3rd column of the 3rd dimension")
x (2, 3, 3)
disp ("Elements in the 3rd row of the 2nd dimension")
x (3, : , 2 )
disp ("Elements in the 2nd row of the 3rd dimension")
 x ( : , 2,3)
disp ("Delete 3rd dimension and change it")
x ( : , :, 3)= [3 2 2; 3 4 5; 1 2 4]
```

Output:

```
3 -    x (end, end, 2) = 0
```

Command Window

New to MATLAB? See resources for Getting Started.

```
x(:,:,3) =

     3     2     2
     1     3     1
     6     4     9

Element in the Ist row of the 2nd column of the 3rd dimension

ans =

     2

Element in the 2nd row of the 3rd column of the 3rd dimension

ans =

     1

Elements in the 3rd row of the 2nd dimension
```

4. Access the element in the 2nd row 3rd column of the 3rd dimension.

Code:

MATLAB

Editor - C:\Users\Prasad\Desktop\MATLAB\demoo.m

demo.m | demoo.m

```
x= [3 25 1; 23 8 5; 8 8 23]
disp ("2 dimensions")
x (end, end, 2) = 0
disp ("Cat function to add 3rd dimension")
x=cat (3, x, [3 2 2; 1 3 1; 6 4 9])
disp ("Element in the Ist row of the 2nd column of the 3rd dimension")
x (1, 2, 3)
disp ("Element in the 2nd row of the 3rd column of the 3rd dimension")
x (2, 3, 3)
disp ("Elements in the 3rd row of the 2nd dimension")
x (3, : , 2 )
disp ("Elements in the 2nd row of the 3rd dimension")
 x ( : , 2,3)
disp ("Delete 3rd dimension and change it")
x ( : , :, 3)= [3 2 2; 3 4 5; 1 2 4]
```

Output:

Command Window

New to MATLAB? See resources for Getting Started.

```
x(:,:,3) =

     3     2     2
     1     3     1
     6     4     9

Element in the Ist row of the 2nd column of the 3rd dimension

ans =

     2

Element in the 2nd row of the 3rd column of the 3rd dimension

ans =

     1

Elements in the 3rd row of the 2nd dimension
```

5. Access all the elements in the 3rd row of 2nd dimension.

Code:

Editor - C:\Users\Prasad\Desktop\MATLAB\demoo.m

demo.m | demoo.m

```
x= [3 25 1; 23 8 5; 8 8 23]
disp ("2 dimensions")
x (end, end, 2) = 0
disp ("Cat function to add 3rd dimension")
x=cat (3, x, [3 2 2; 1 3 1; 6 4 9])
disp ("Element in the Ist row of the 2nd column of the 3rd dimension")
x (1, 2, 3)
disp ("Element in the 2nd row of the 3rd column of the 3rd dimension")
x (2, 3, 3)
disp ("Elements in the 3rd row of the 2nd dimension")
x (3, : , 2 )
disp ("Elements in the 2nd row of the 3rd dimension")
 x ( : , 2,3)
disp ("Delete 3rd dimension and change it")
x ( : , :, 3)= [3 2 2; 3 4 5; 1 2 4]
```

Output:

Command Window

New to MATLAB? See resources for Getting Started.

```
Element in the 2nd row of the 3rd column of the 3r

ans =

     1

Elements in the 3rd row of the 2nd dimension

ans =

     0     0     0

Elements in the 2nd row of the 3rd dimension

ans =

     2
     3
     4
```

6. Access all the elements of the 2nd column 3rd dimension.

Code:

Editor - C:\Users\Prasad\Desktop\MATLAB\demoo.m

demo.m | demoo.m

```
x= [3 25 1; 23 8 5; 8 8 23]
disp ("2 dimensions")
x (end, end, 2) = 0
disp ("Cat function to add 3rd dimension")
x=cat (3, x, [3 2 2; 1 3 1; 6 4 9])
disp ("Element in the Ist row of the 2nd column of the 3rd dimension")
x (1, 2, 3)
disp ("Element in the 2nd row of the 3rd column of the 3rd dimension")
x (2, 3, 3)
disp ("Elements in the 3rd row of the 2nd dimension")
x (3, : , 2 )
disp ("Elements in the 2nd row of the 3rd dimension")
 x ( : , 2,3)
disp ("Delete 3rd dimension and change it")
x ( : , :, 3)= [3 2 2; 3 4 5; 1 2 4]
```

Output:

Command Window

New to MATLAB? See resources for Getting Started.

```
Elements in the 2nd row of the 3rd dimension

ans =

     2
     3
     4

Delete 3rd dimension and change it

x(:,:,1) =

     3    25     1
    23     8     5
     8     8    23


x(:,:,2) =
```

7. Delete dimension 3 and change it to [2 5 3; 8 9 0; 6 5 4]

Code

Editor - C:\Users\Prasad\Desktop\MATLAB\demoo.m

demo.m | demoo.m

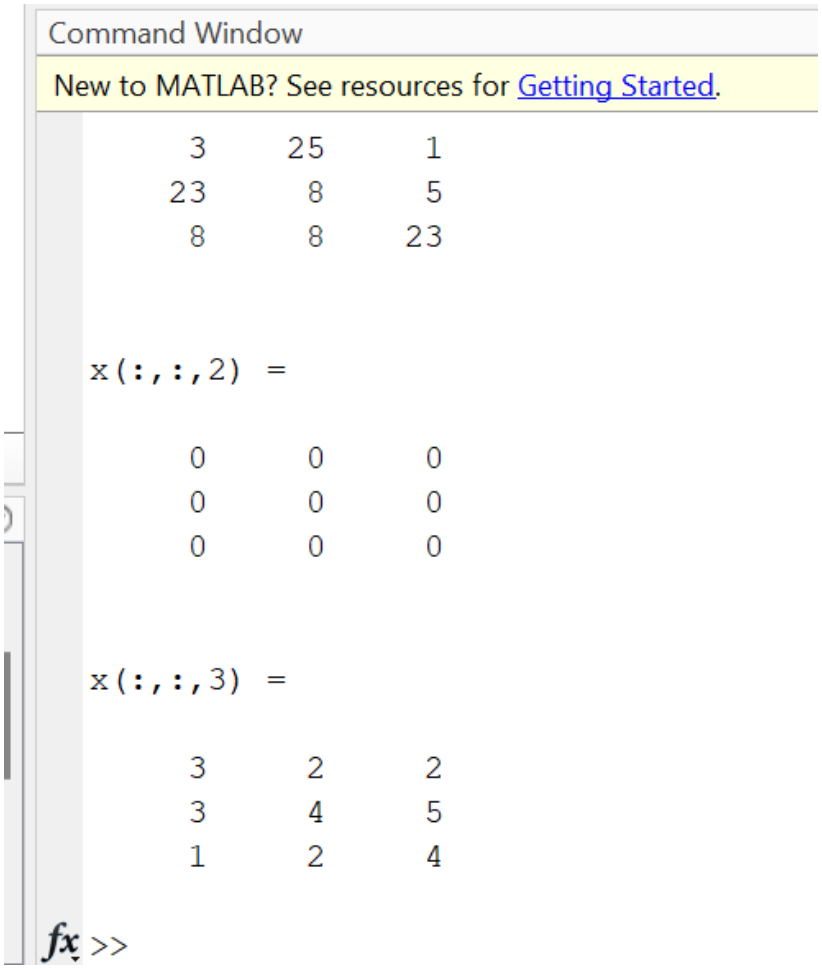
```
x= [3 25 1; 23 8 5; 8 8 23]
disp ("2 dimensions")
x (end, end, 2) = 0
disp ("Cat function to add 3rd dimension")
x=cat (3, x, [3 2 2; 1 3 1; 6 4 9])
disp ("Element in the Ist row of the 2nd column of the 3rd dimension")
x (1, 2, 3)
disp ("Element in the 2nd row of the 3rd column of the 3rd dimension")
x (2, 3, 3)
disp ("Elements in the 3rd row of the 2nd dimension")
x (3, : , 2 )
disp ("Elements in the 2nd row of the 3rd dimension")
 x ( : , 2,3)
disp ("Delete 3rd dimension and change it")
x ( : , :, 3)= [3 2 2; 3 4 5; 1 2 4]
```

Output:

Command Window

New to MATLAB? See resources for Getting Started.

```
     4

Delete 3rd dimension and change it

x(:,:,1) =

     3    25     1
    23     8     5
     8     8    23


x(:,:,2) =

     0     0     0
     0     0     0
     0     0     0


x(:,:,3) =
```


Command Window
New to MATLAB? See resources for Getting Started.
3 25 1
23 8 5
8 8 23
x(:,:,2) =
0 0 0
0 0 0
0 0 0
x(:,:,3) =
3 2 2
3 4 5
1 2 4
>>