**Image Processing**

**(01CE0507)**

**Department of Computer Engineering**

**5th Semester**

**Lab Manual**

**(July-Dec 2022)**

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab** | **Programs** | **Date** | **Signature** |
| 1 | Study of matlab image processing toolkit and various commands on matlab. |  |  |
| 2 | Point processing in spatial domain   1. Negation of an image 2. Thresholding of an image 3. Contrast Stretching of an image |  |  |
| 3 | Write a program for histogram equalization. |  |  |
| 4 | Write a program to apply various filtering techniques in matlab.   1. Low pass filtering 2. High pass filtering 3. Median filtering |  |  |
| 5 | Write a program for image segmentation   1. Local thresholding 2. Global thresholding |  |  |
| 6 | Write a program for color image processing   1. Color approximation 2. Quantization |  |  |
| 7 | Write a program, for Image restoration   1. Facial Images 2. Texture Images |  |  |
| 8 | Write a program for edge detection. |  |  |
| 9 | Write a program for smoothening and sharpening for 8-bit color image. |  |  |
| 10 | Write a program to implement morphological operations. |  |  |

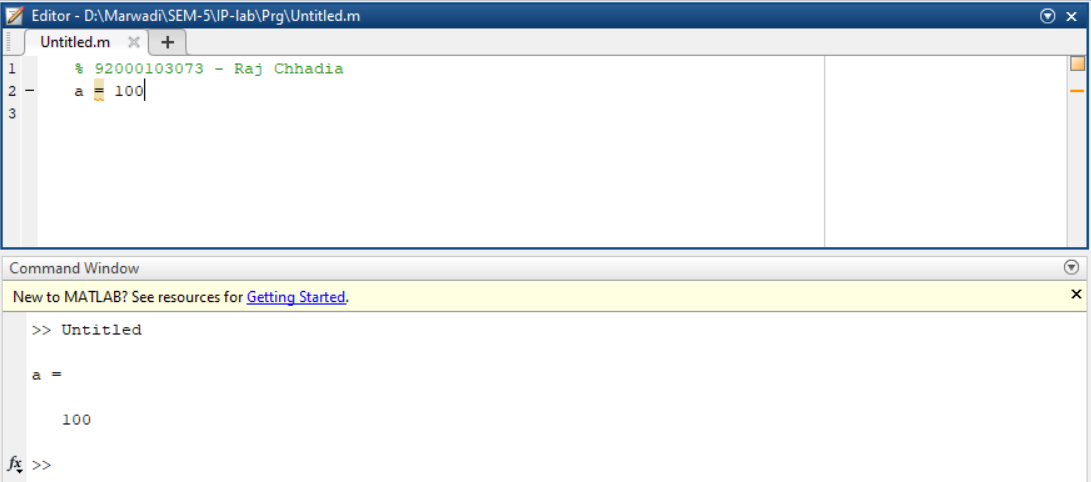
**Practical 1**

**Aim:** Study of matlab image processing toolkit and various commands on matlab.

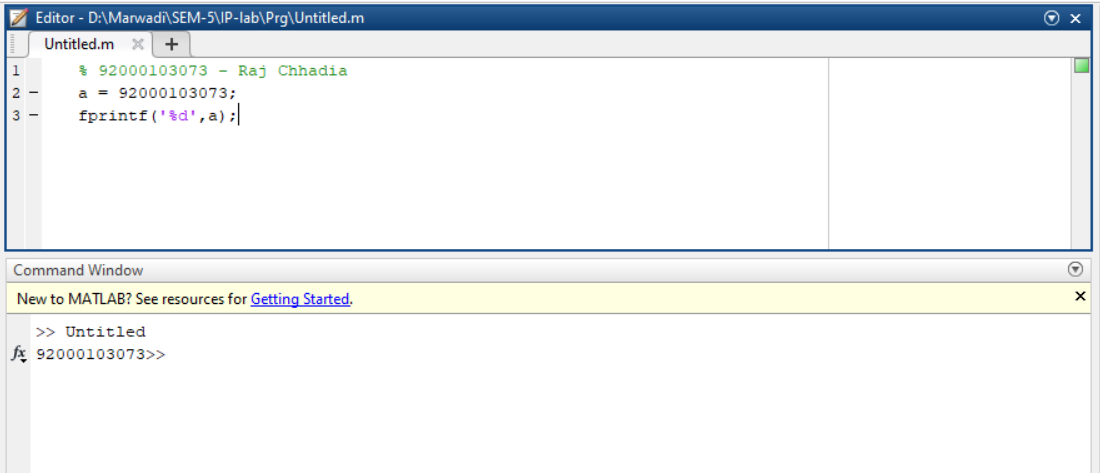
* **Introduction to Various component of MATLAB Tool like editor, command window, workspace.**
* **Basic Syntax, Variables, Commands**
  + **Use of Semicolon (;) in MATLAB**

1. **Without Semicolon**

* Use semicolons to separate rows in an array creation command, or to suppress the output display of a line of code.



1. **With Semicolon**



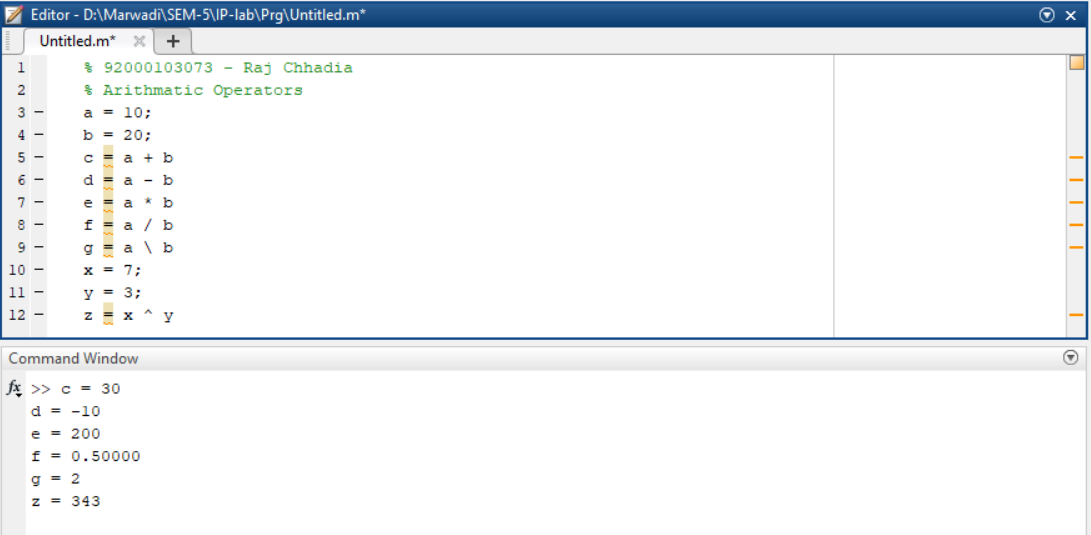
* + **Adding Comments**
* Comments are non-executable text within the body of a program. They can be declared by ‘%’



* + **Commonly used Operators and Special Characters**

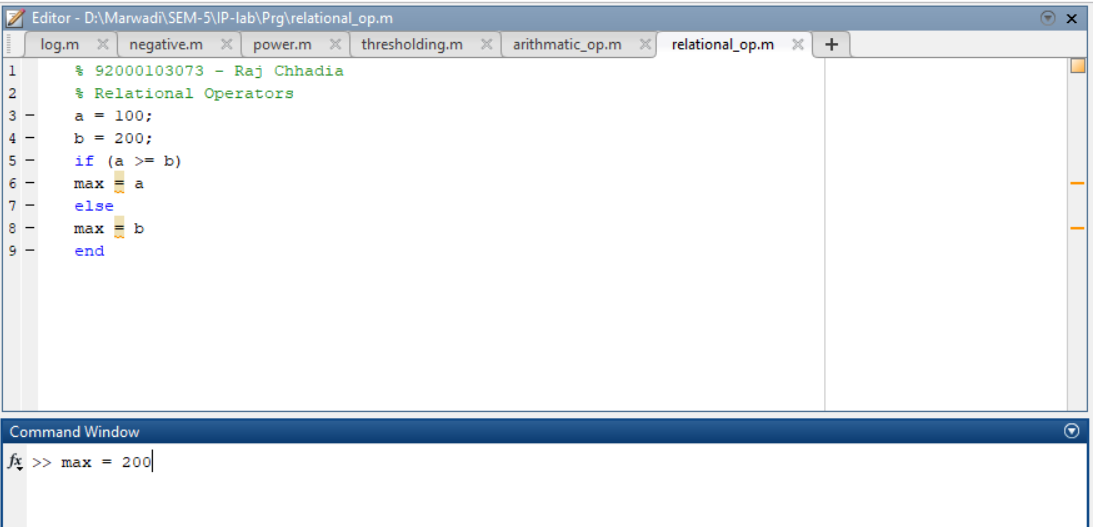
1. **Arithmatic Operation**

|  |  |  |
| --- | --- | --- |
| **No.** | **Symbol** | **Description** |
|  | **+** | Addition or unary plus. A+B adds the values stored in variables A and B. |
|  | **-** | Subtraction or unary minus. A-B subtracts value of B from A. |
|  | **\*** | Matrix multiplication. C = A\*B is the linear algebraic product of the matrices A and B. |
|  | **.\*** | Array multiplication. A.\*B is the element-by-element product of the arrays A and B. |
|  | **/** | Slash or matrix right division. B/A is roughly the same as B\*inv(A). |
|  | **./** | Array right division. A./B is the matrix with elements A(i,j)/B(i,j). |
|  | **\** | Backslash or matrix left division. If A is a square matrix, A\B is roughly the same as inv(A)\*B. |
|  | **.\** | Array left division. A.\B is the matrix with elements B(i,j)/A(i,j). |
|  | **^** | Matrix power. X^p is X to the power p, if p is a scalar. If p is an integer, the power is computed by repeated squaring. |
|  | **.^** | Array power. A.^B is the matrix with elements A(i,j) to the B(i,j) power. |
|  | **‘** | Matrix transpose. A' is the linear algebraic transpose of A. |
|  | **.’** | Array transpose. A.' is the array transpose of A. |



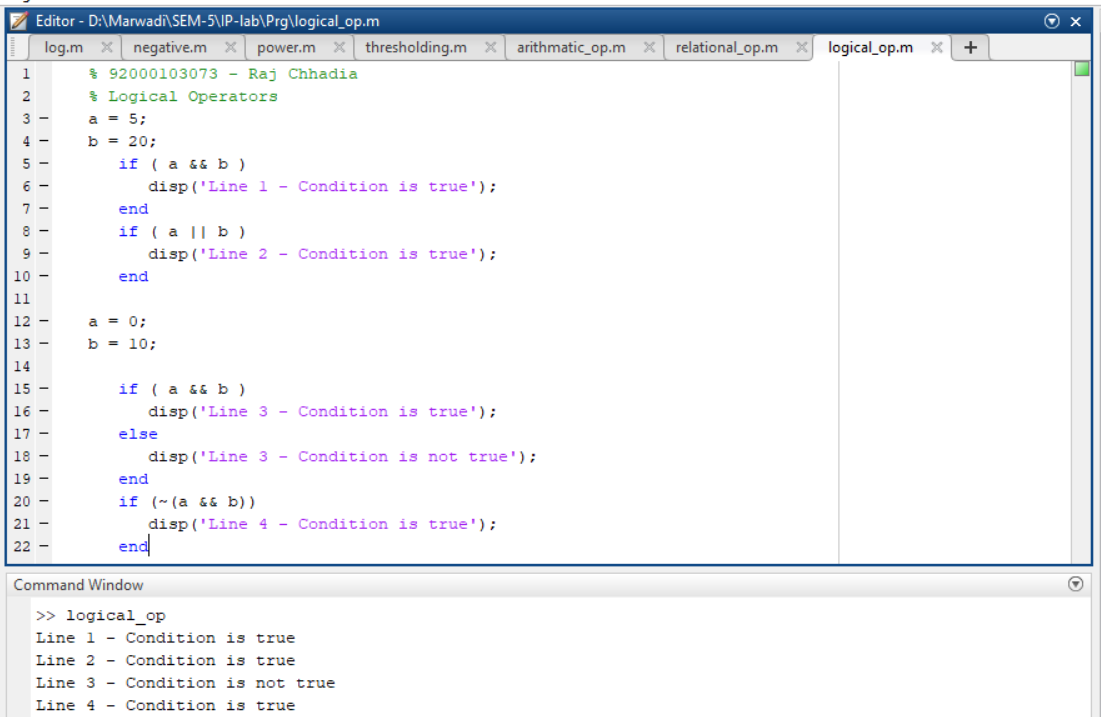
1. **Relation Operation**

|  |  |
| --- | --- |
| **Symbol** | **Description** |
| == | Determine Equality |
| >= | Determine greater than or equal to |
| > | Determine greater than |
| <= | Determine less than or equal to |
| < | Determine less than |
| ~= | Determine inequality |
| isequal | Determine array equality. |
| isequaln | Determine array equality, treating NaN values as equal |

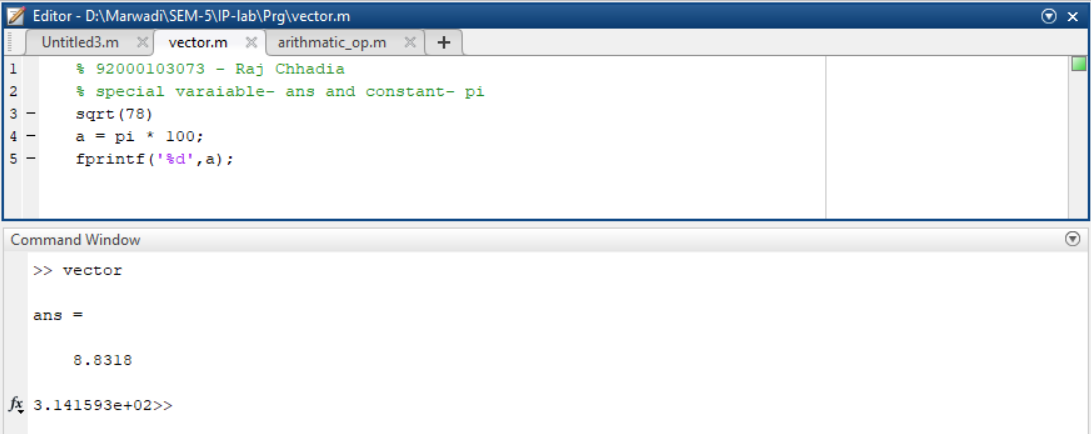


1. **Logical opereation**

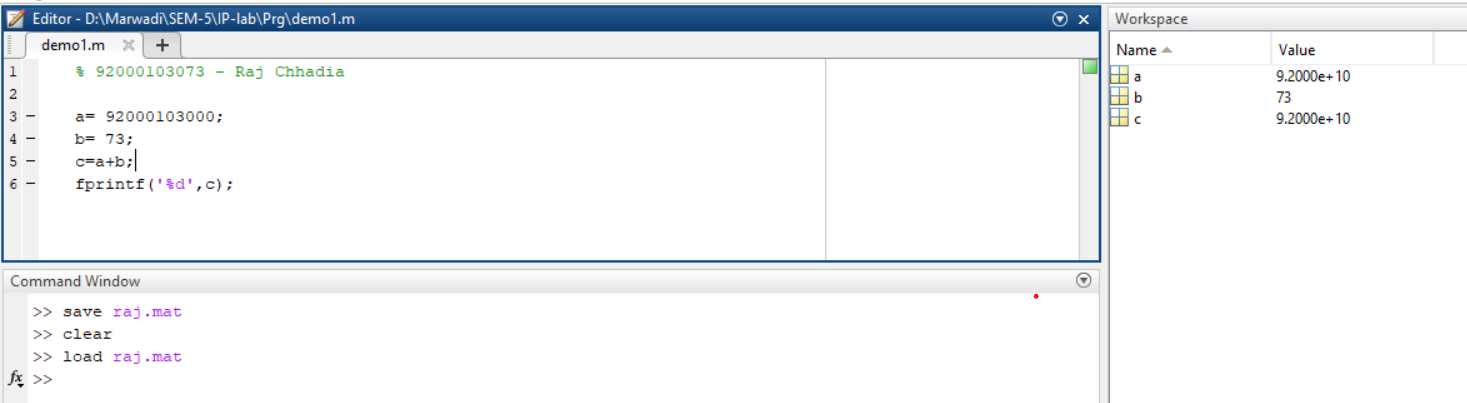
|  |  |
| --- | --- |
| **Symbol** | **Description** |
| && , || | Short Circuit Operations |
| & | Logical AND |
| ~ | Logical NOT |
| | | Logical OR |
| xor | Logical XOR |
| all | Determine if all array elements are nonzero or not |
| any | Determine if any value is zero in array |
| false | Logical 0 (low) |
| true | Logical 1(high) |

****

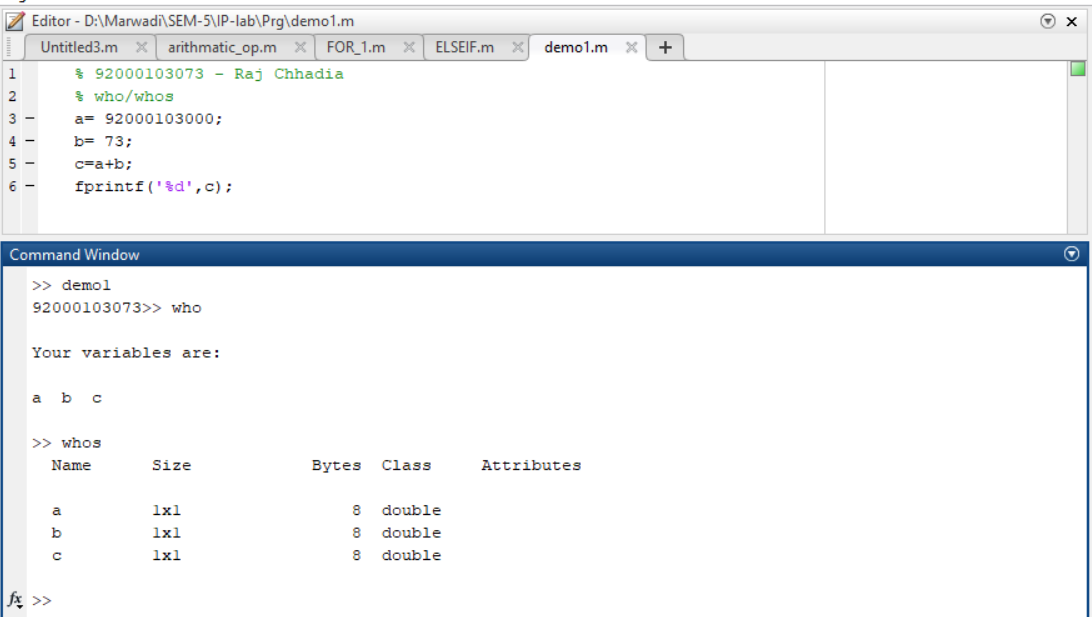
* + **Special Variables and Constants**
* sqrt, pi are some of the special constants and ans is one of the special variables to store the answer

****

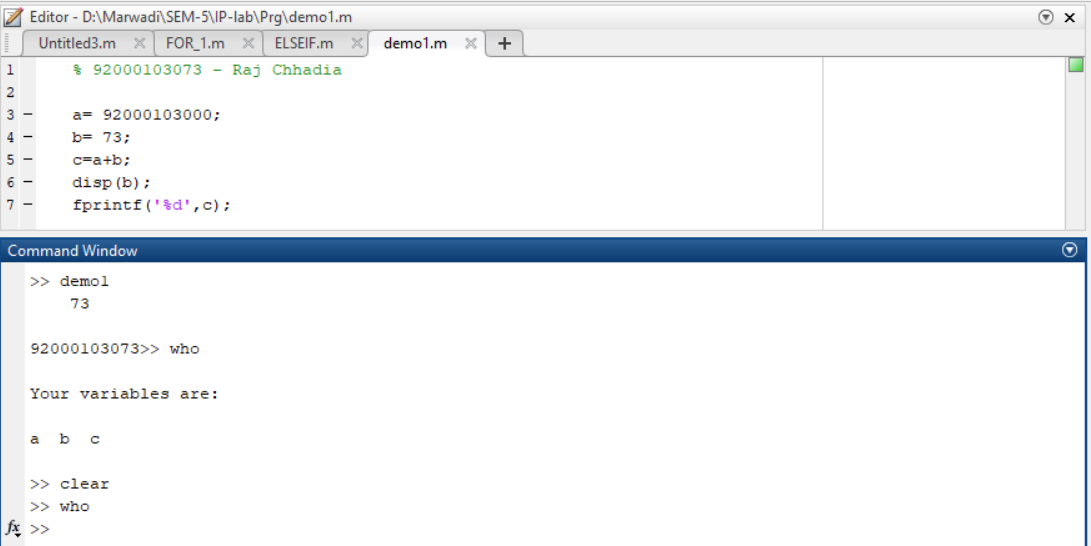
* + **save and load Command**
* save command saves workspace variables in a file. load command loads workspace variables from a file.

****

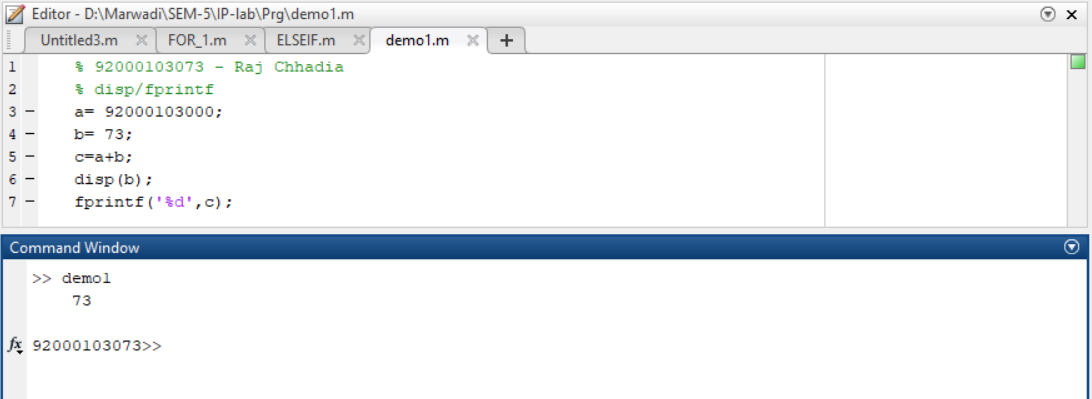
* + **who and whos command**

****

* + **clear and clc Command**
* clc clears command window. clear removes variables from memory.

****

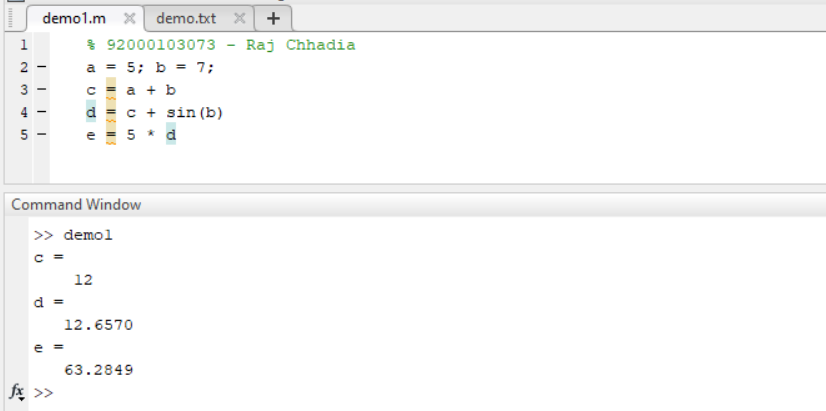
* + **disp and fprintf command**
* disp displays contents of an array or string. fprintf performs formatted writes to screen or file.

****

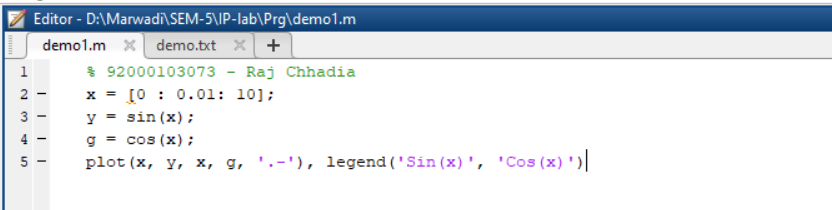
* + **input command**

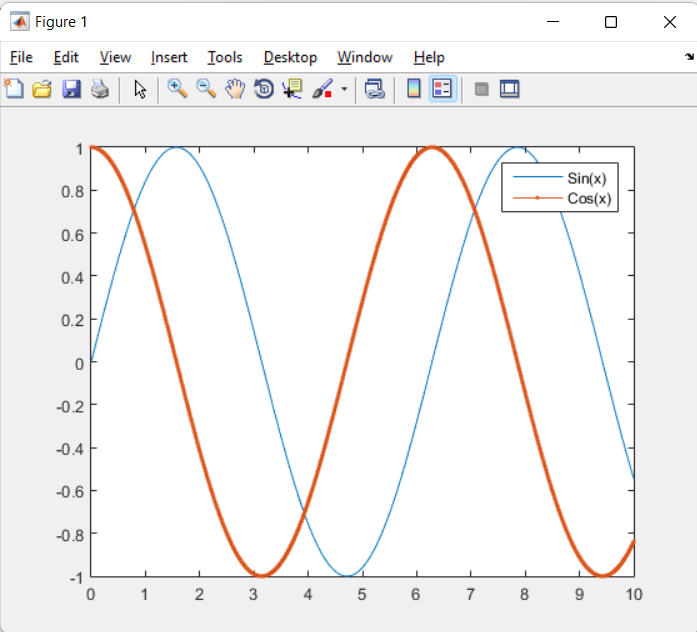
****

* + **M-Files**
* Script files are program files with .m extension. In these files, you write series of commands, which you want to execute together.

****

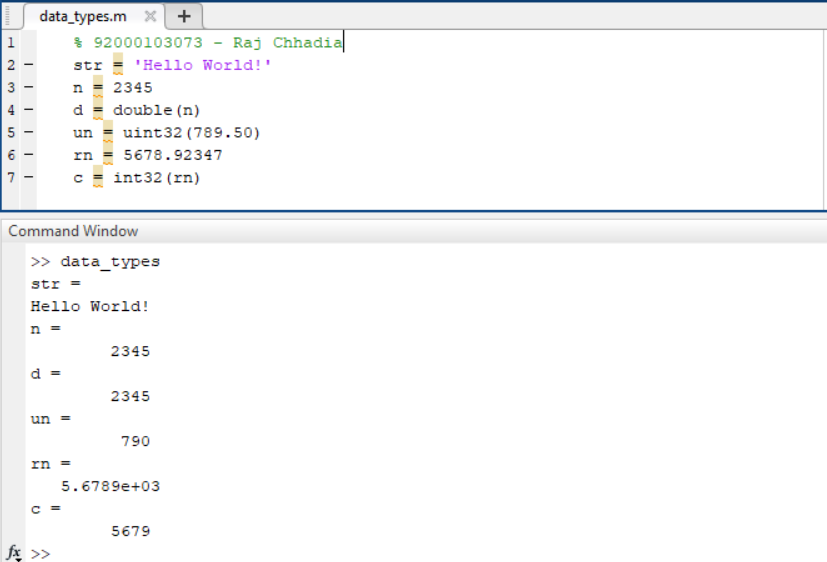
* + **Plotting Commands**
* plot command is used to plot a graph or curve.

****



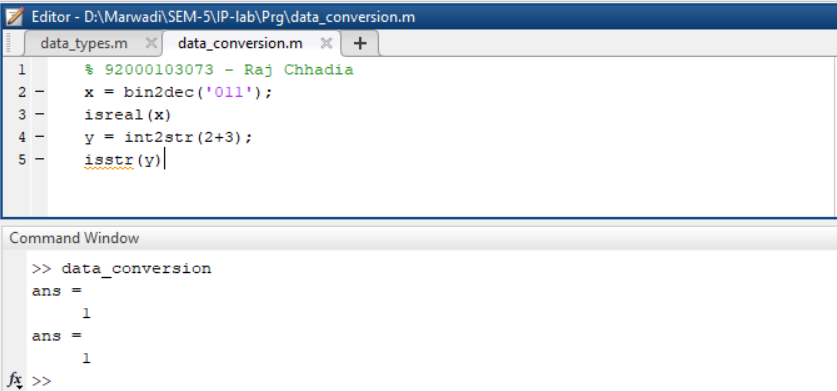
* **Data Types in MATLAB**
  + **List of Data Types**

|  |  |
| --- | --- |
| **No.** | **Data Types** |
|  | int8 |
|  | uint8 |
|  | int32 |
|  | uint32 |
|  | int64 |
|  | uint64 |
|  | double |
|  | char |
|  | cell array |
|  | structure |

****

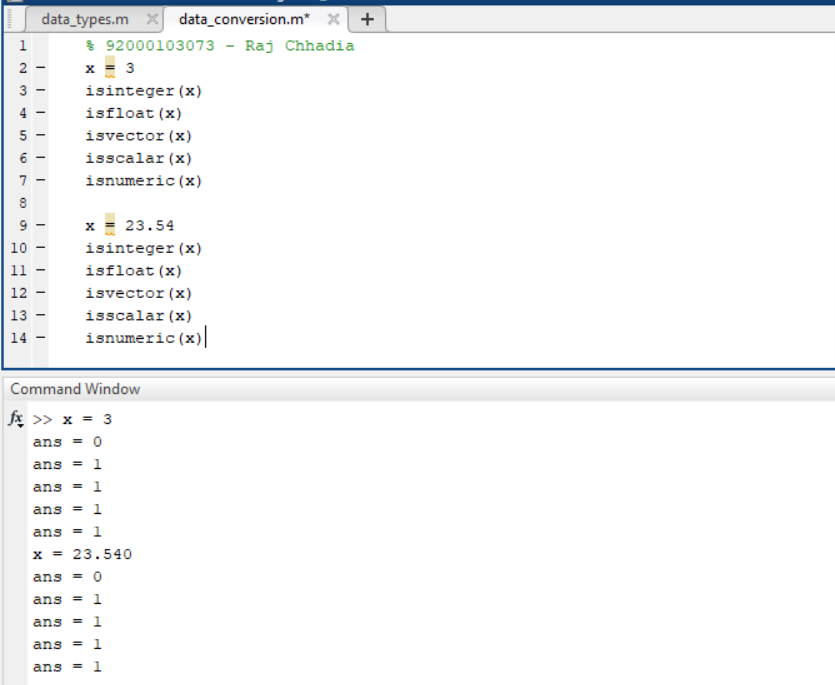
* + **Data Type Conversion**

|  |  |
| --- | --- |
| **Function** | **Purpose** |
| char | Convert to character array (string) |
| int2str | Convert integer data to string |
| num2str | Convert number to string |
| str2double | Convert string to double-precision value |
| str2num | Convert string to number |
| bin2dec | Convert binary number to decimal number |
| dec2base | Convert decimal to base N number in string |
| dec2bin | Convert decimal to binary number in string |
| hex2dec | Convert hexadecimal number string to decimal number |
| mat2cell | Convert array to cell array with potentially different sized cells |
| cellstr | Create cell array of strings from character array |

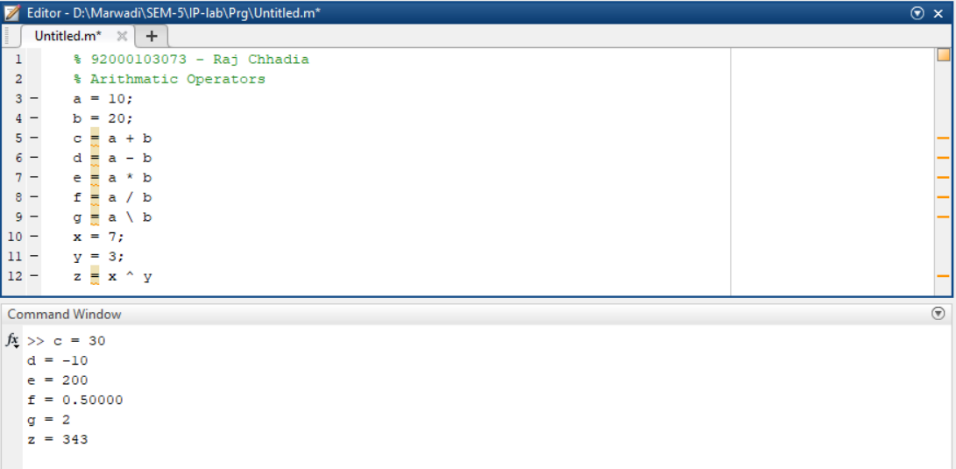


* + **Determination of Data Types**

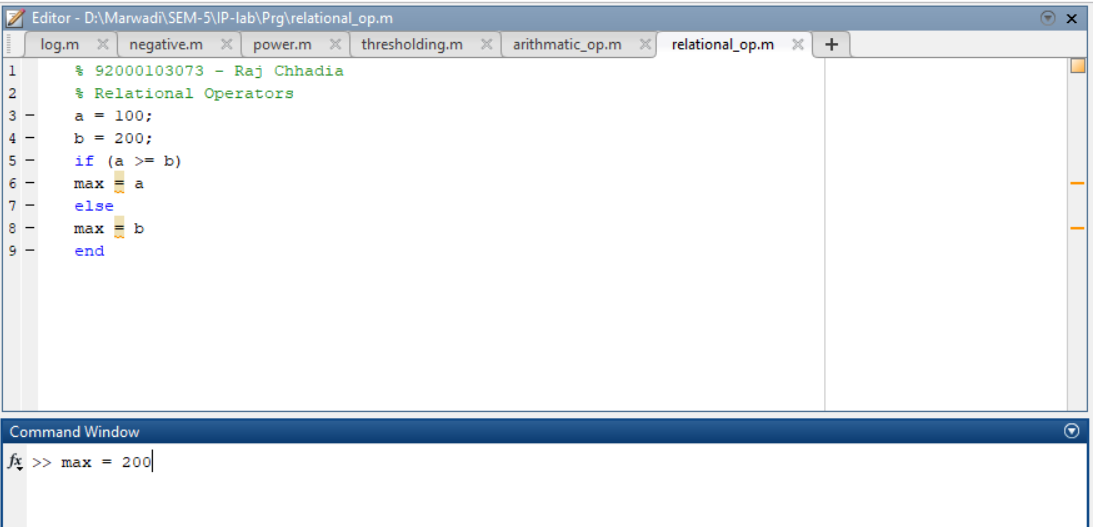
|  |  |
| --- | --- |
| **Function** | **Purpose** |
| isinteger | Determine if input is integer |
| isfloat | Determine if input is float |
| isvector | Determine if input is vector |
| isscalar | Determine if input is scalar |
| isnumeric | Determine if input is numeric array |



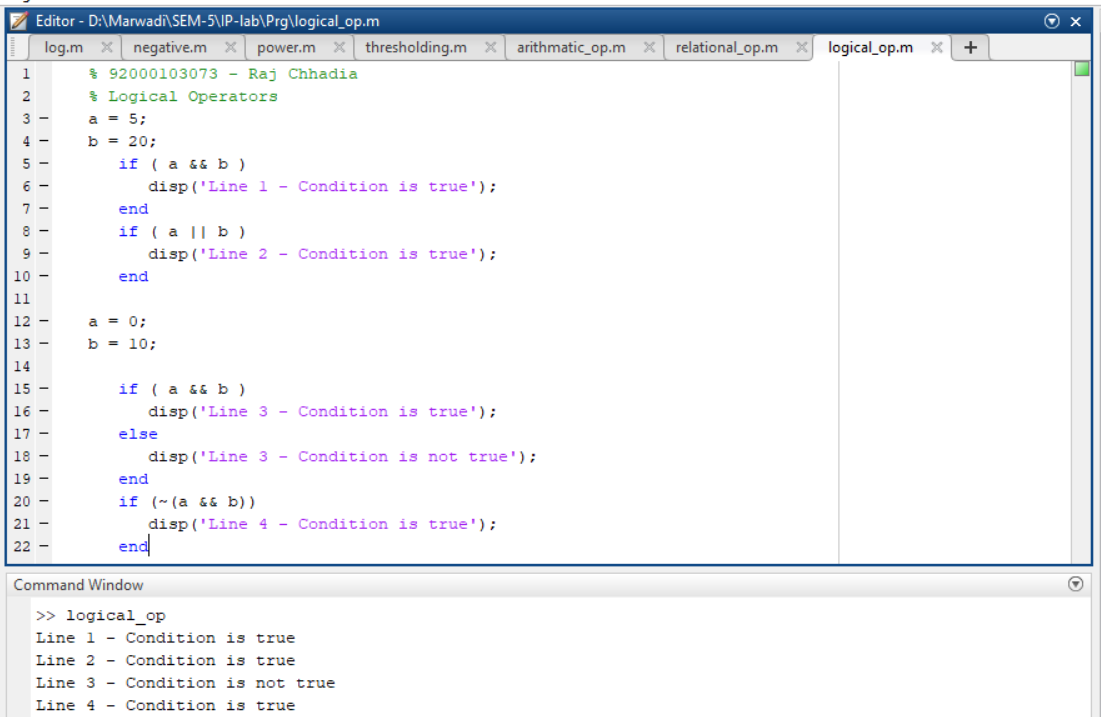
* **Operators in MATLAB**
  + **Types of Operators**
    - **Arithmetic Operators**



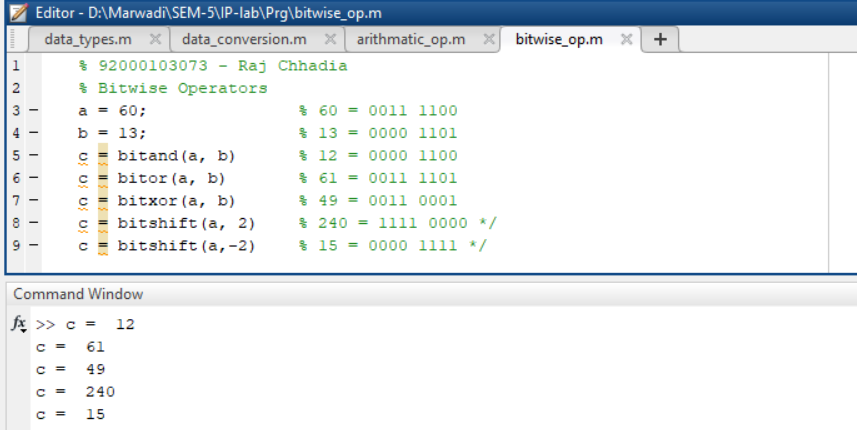
* + - **Relational Operators**



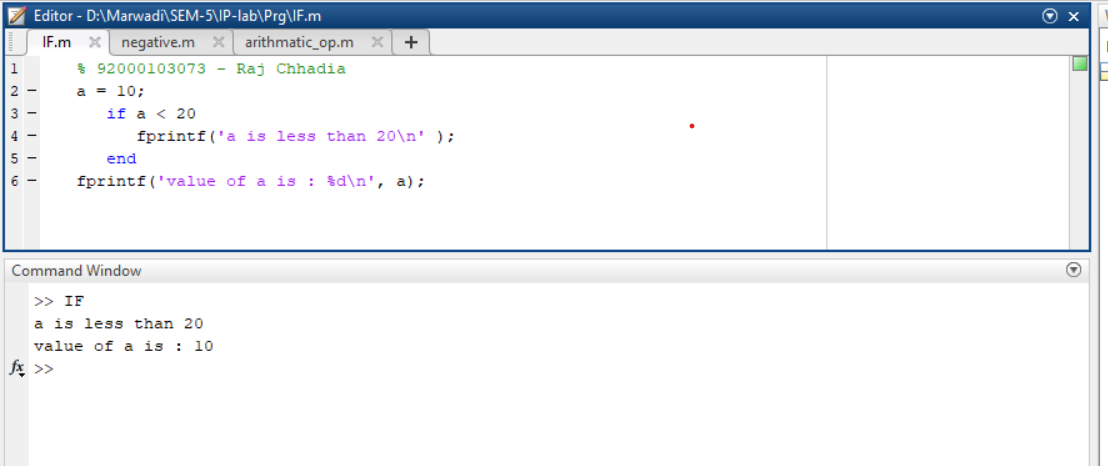
* + - **Logical Operators**



* + - **Bitwise Operations**

****

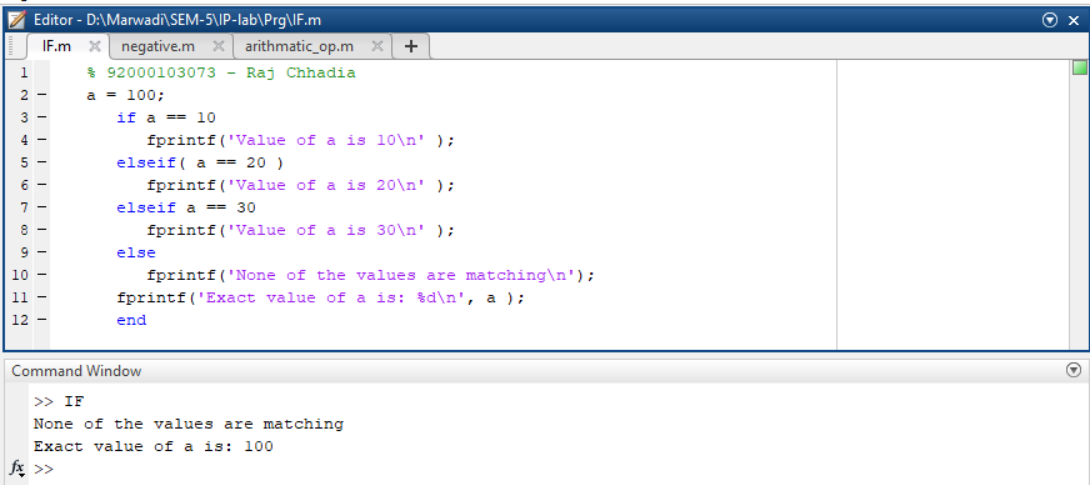
* **Decision Making in MATLAB**
  + **if ... end statement**

****

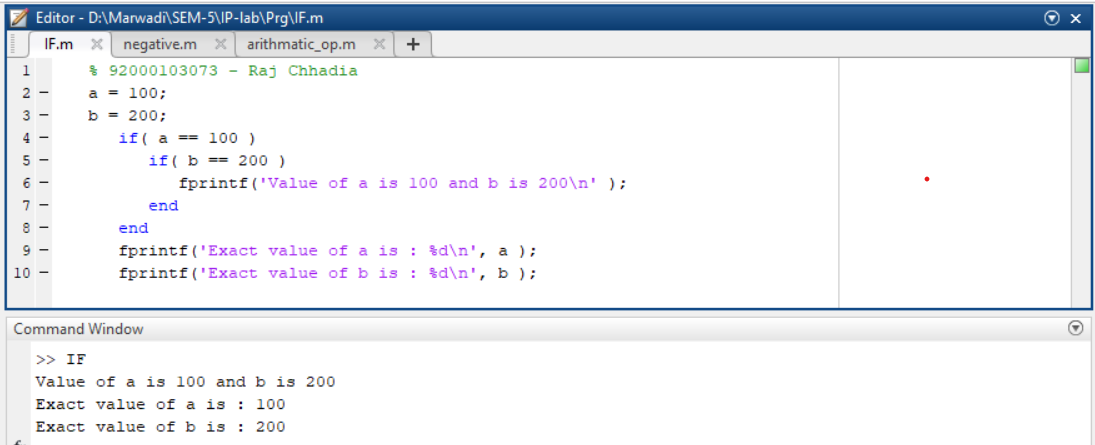
* + **if...else...end statement**

****

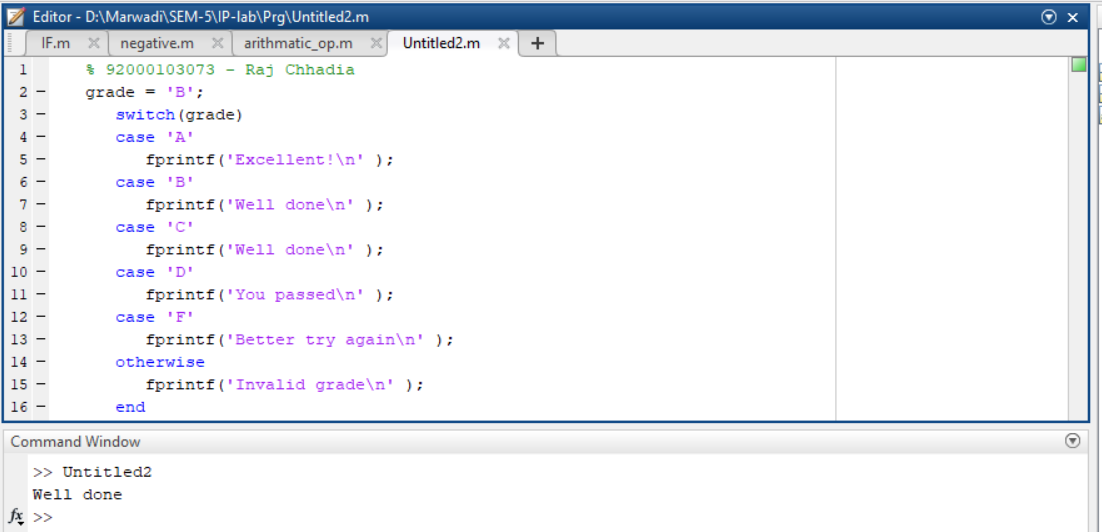
* + **If... elseif...elseif...else...end statements**

****

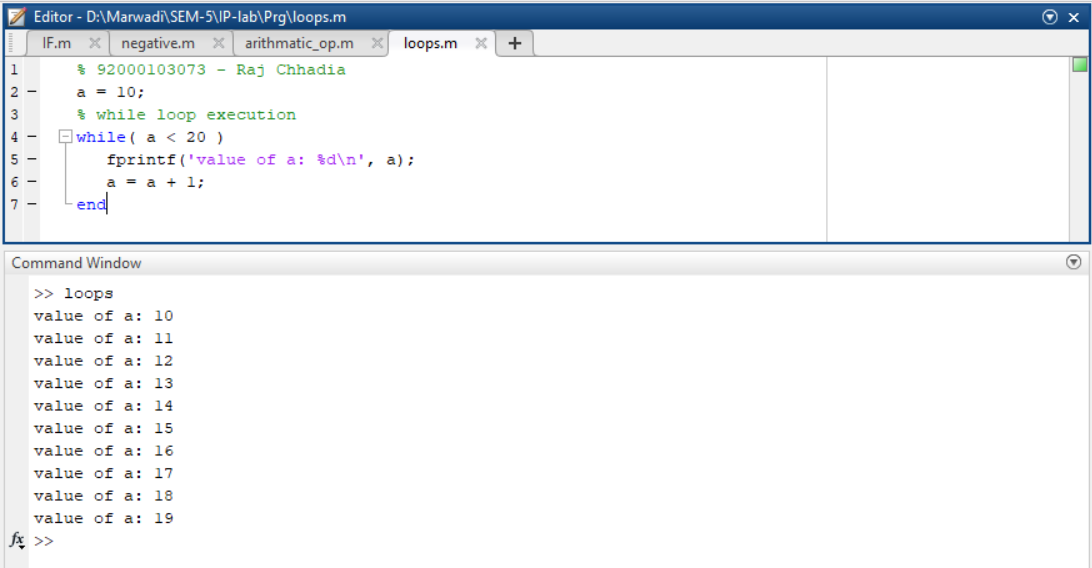
* + **nested if statements**

****

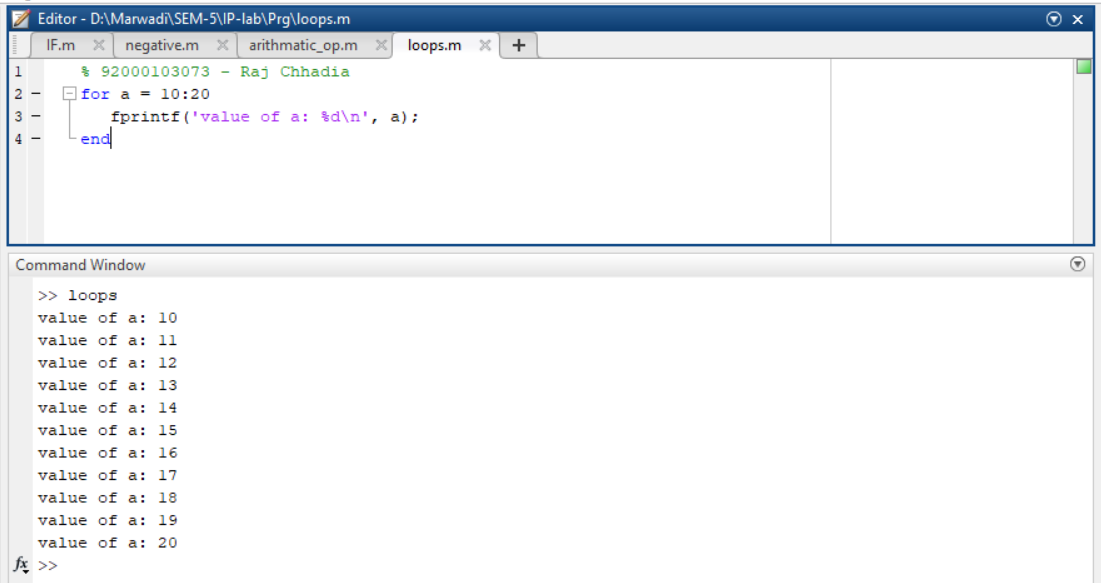
* + **switch statement**

****

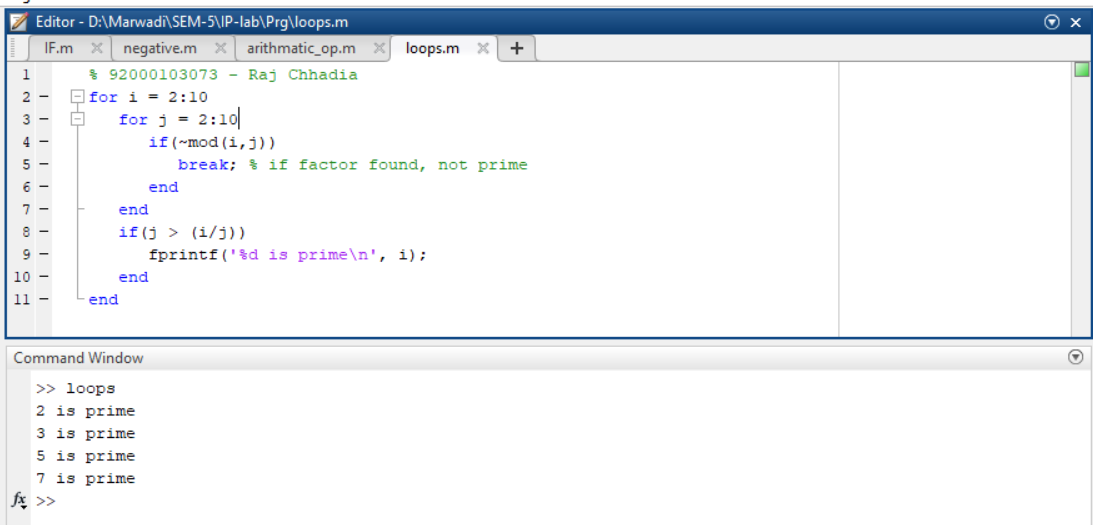
* **Loop Types in MATLAB**
  + **while loop**

****

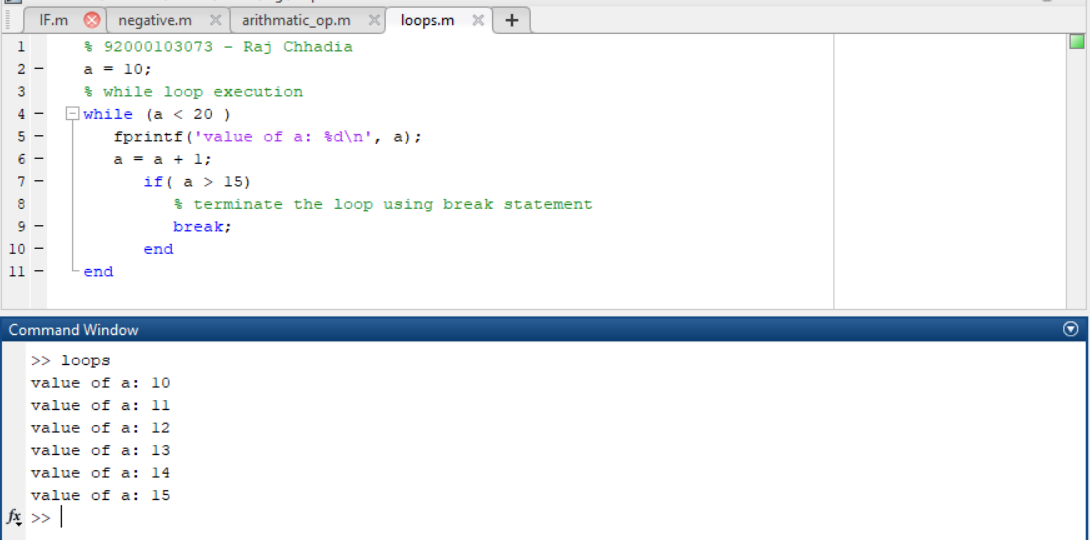
* + **for loop**

****

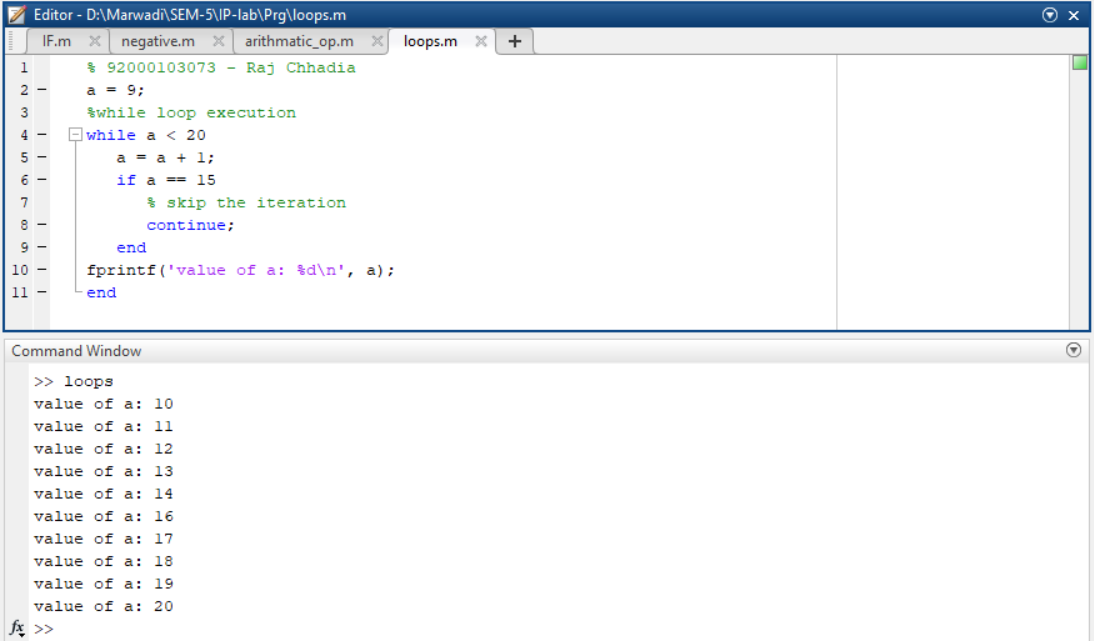
* + **nested loops**

****

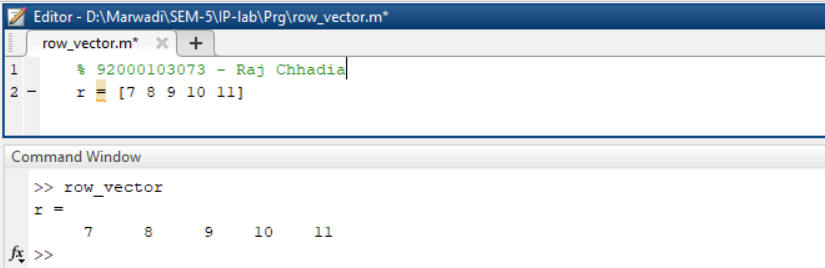
* + **break statement**

****

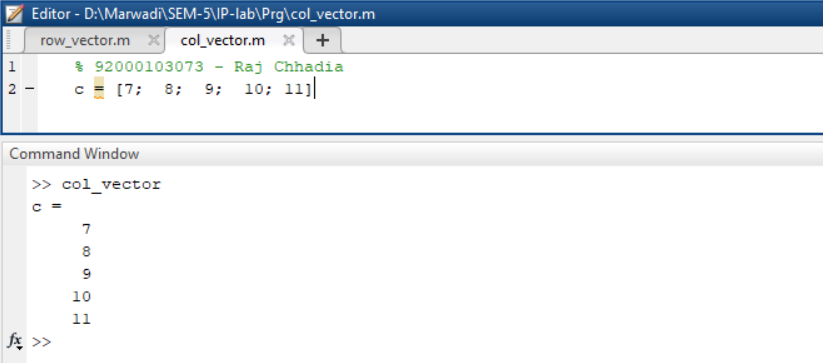
* + **continue statement**

****

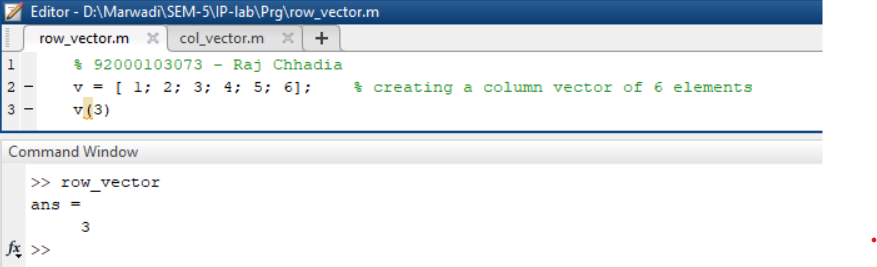
* **Vectors in MATLAB**
  + **Row Vectors**

****

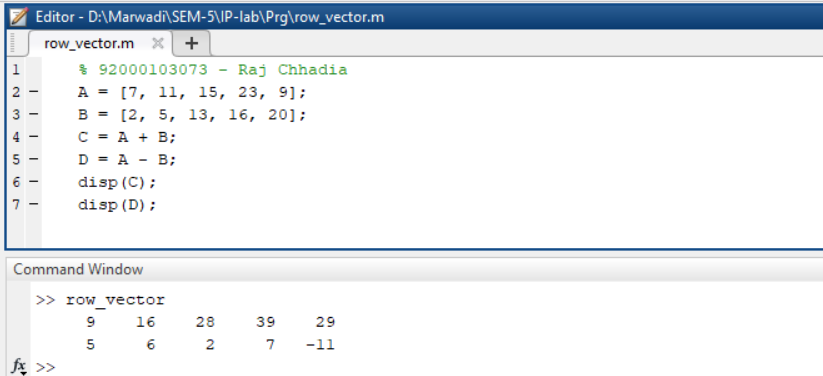
* + **Column Vectors**

****

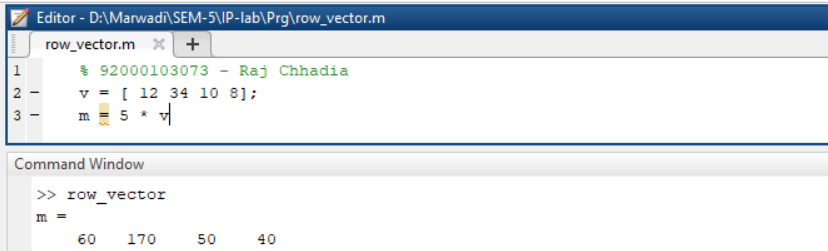
* + **Referencing the Elements of a Vector**

****

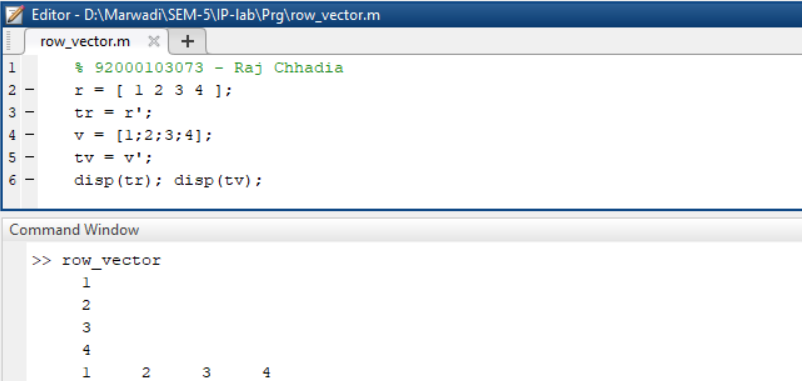
* + **Vector Operations**
    - **Addition and Subtraction of Vectors**

****

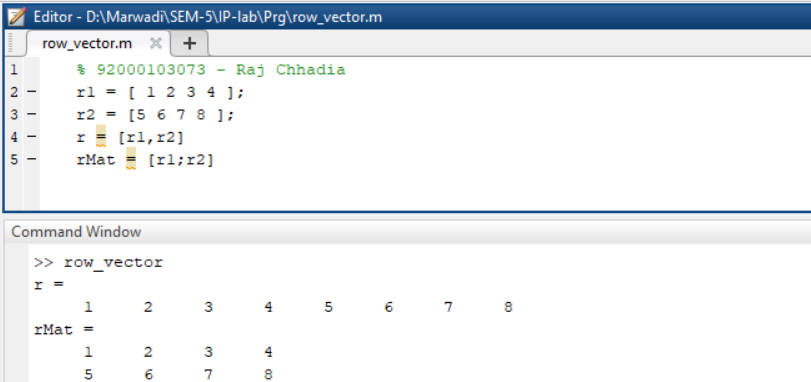
* + - **Scalar Multiplication of Vectors**

****

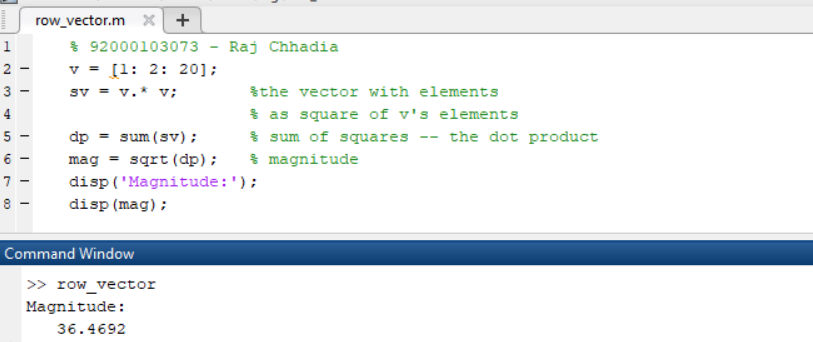
* + - **Transpose of a Vector**



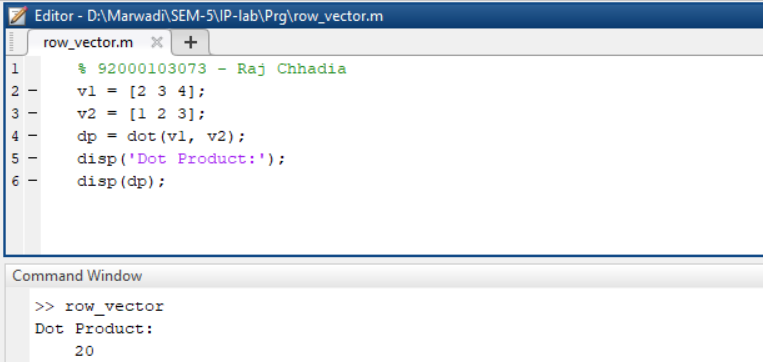
* + - **Appending Vectors**



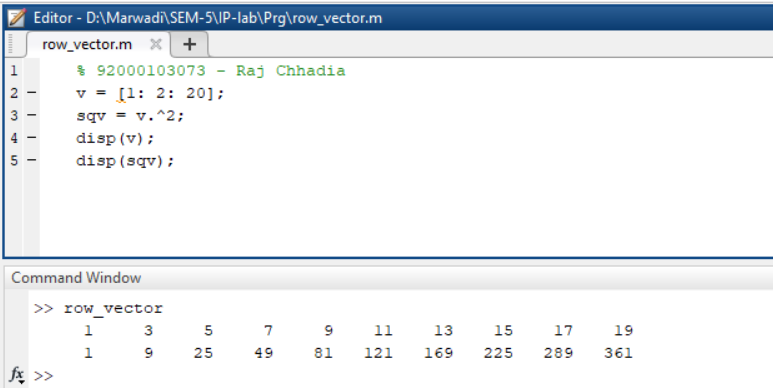
* + - **Magnitude of a Vector**



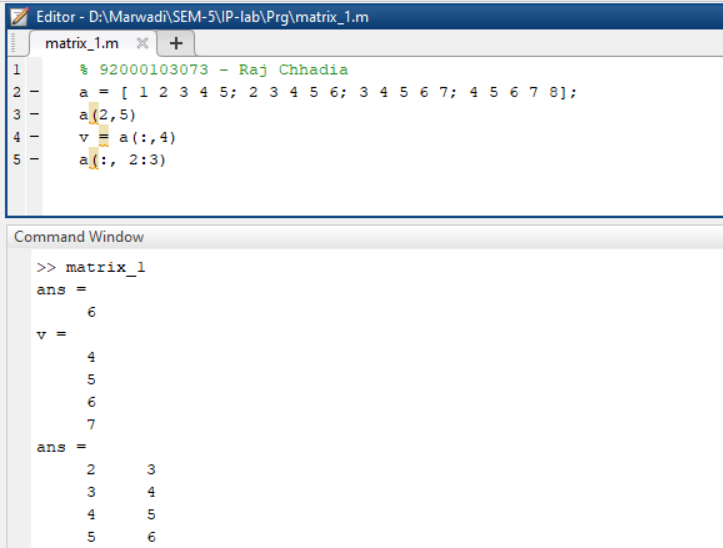
* + - **Vector Dot Product**



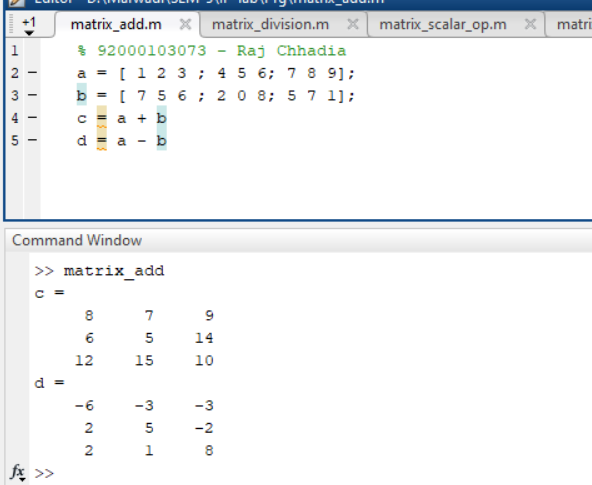
* + - **Vectors with Uniformly Spaced Elements**



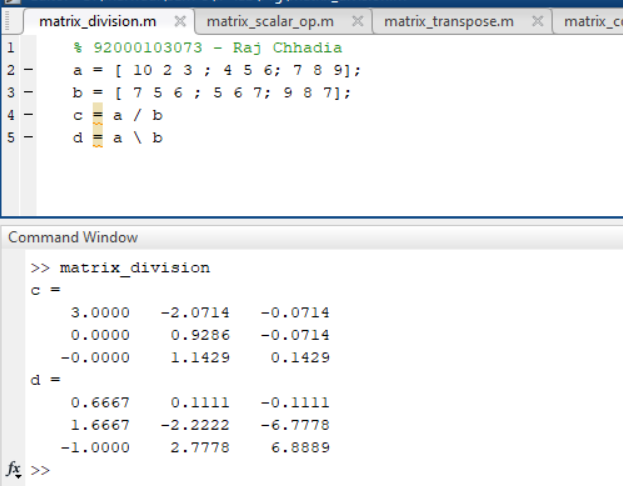
* **Matrix in MATLAB**
  + **Referencing the Elements of a Matrix**
* To reference an element in the mth row and nth column, of a matrix *mx*, we write − mx(m, n);
* To reference all the elements in the mth column we type A(:,m).



* + **Matrix Operations**
    - **Addition and Subtraction of Matrices**



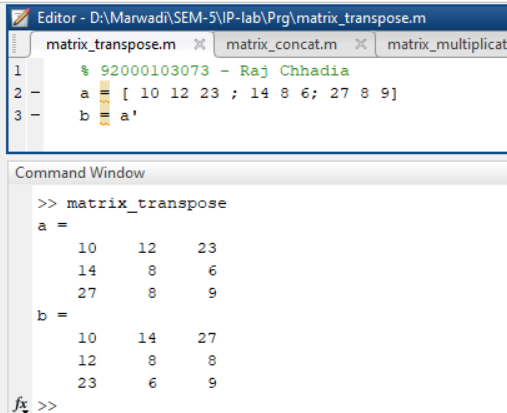
* + - **Division of Matrices**



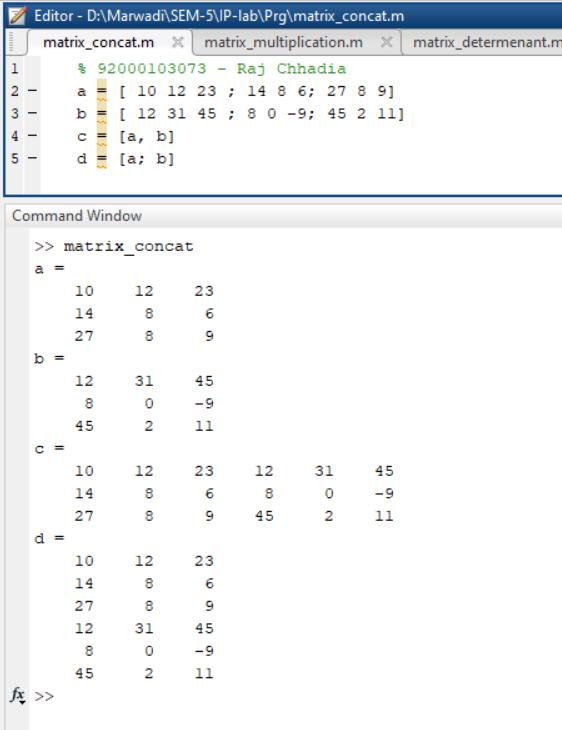
* + - **Scalar Operations of Matrices**



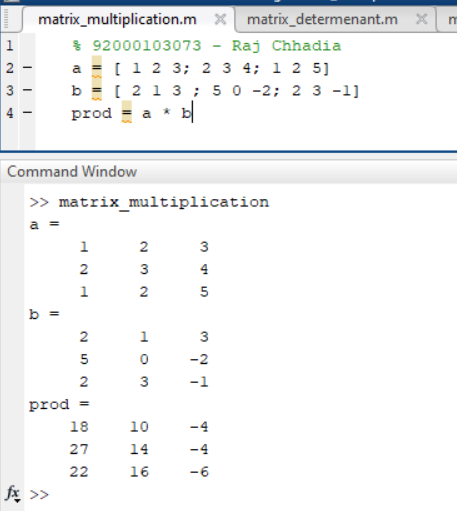
* + - **Transpose of a Matrix**



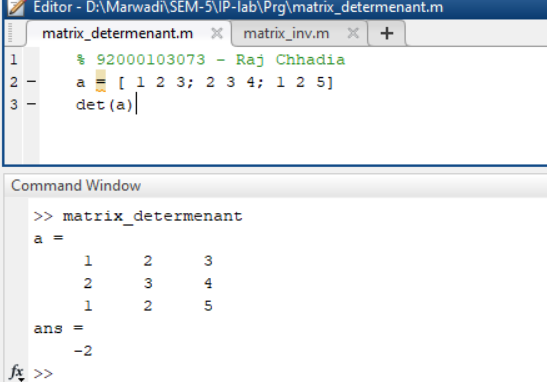
* + - **Concatenating Matrices**



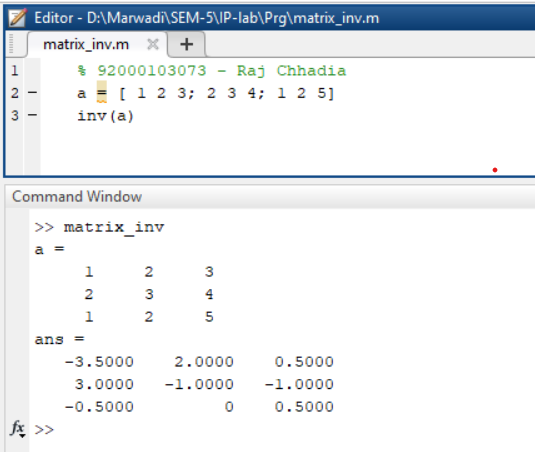
* + - **Matrix Multiplication**



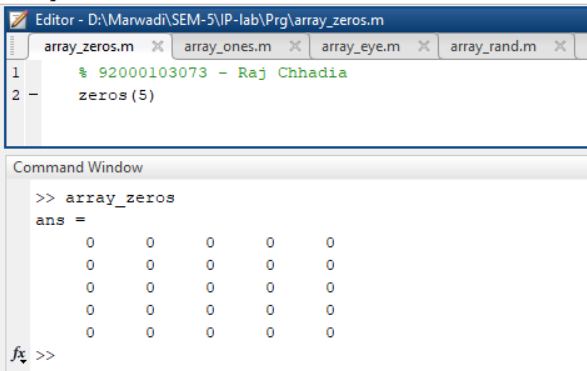
* + - **Determinant of a Matrix**



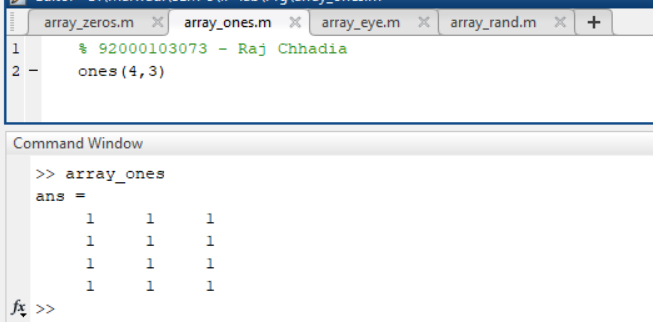
* + - **Inverse of a Matrix**



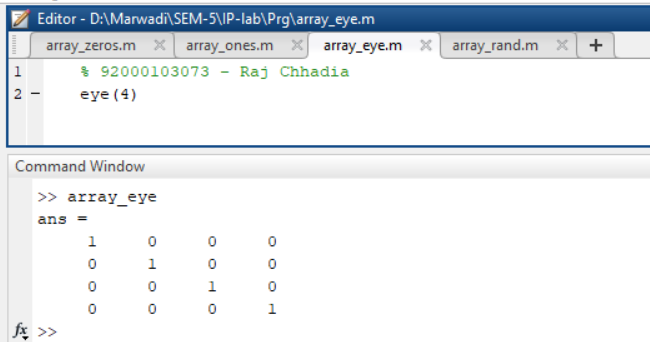
* **Arrays in MATLAB**
  + **Special Arrays in MATLAB**
    - **zeros( ) function**
* The zeros() function creates an array of all zeros



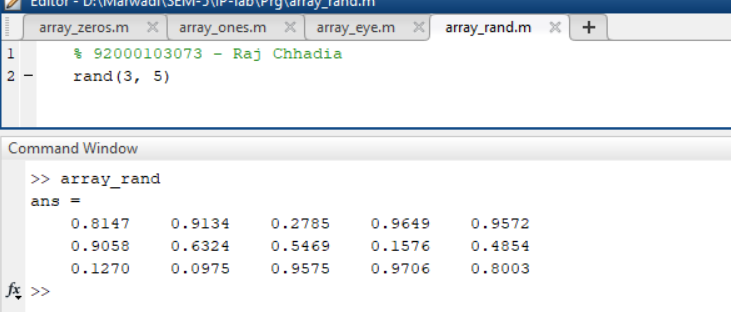
* + - **ones( ) function**
* The ones() function creates an array of all ones



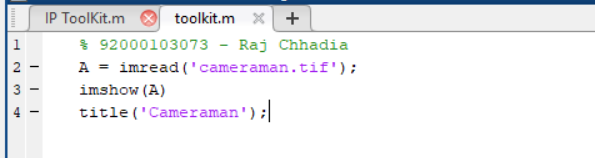
* + - **eye( ) function**
* The eye() function creates an identity matrix.



* + - **rand( ) function**
* The rand() function creates an array of uniformly distributed random numbers on (0,1)

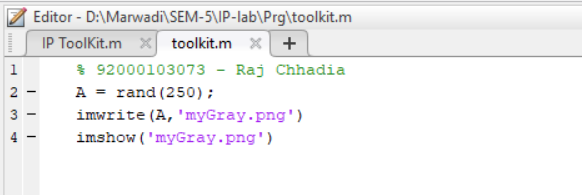


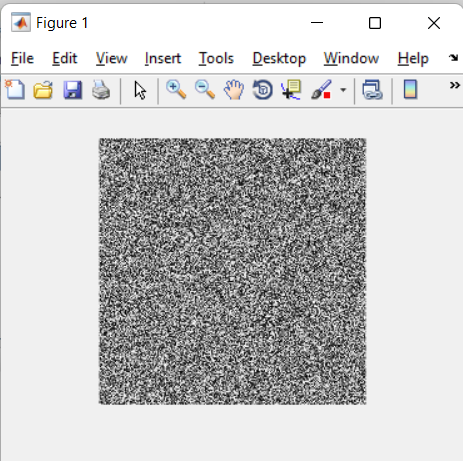
* **Image Processing Toolkit**
  + **imread( ) and imshow( ) command**



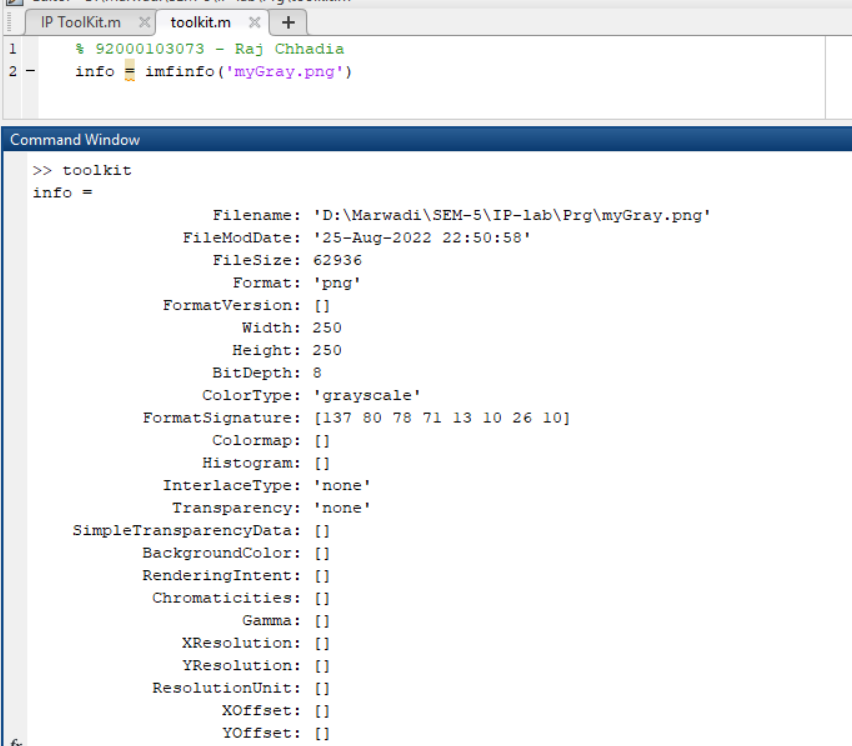


* + **imwrite( ) command**

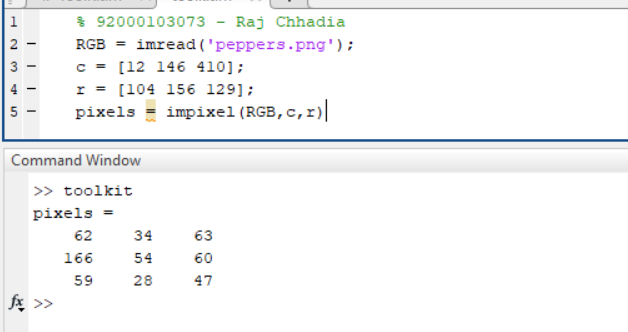


****

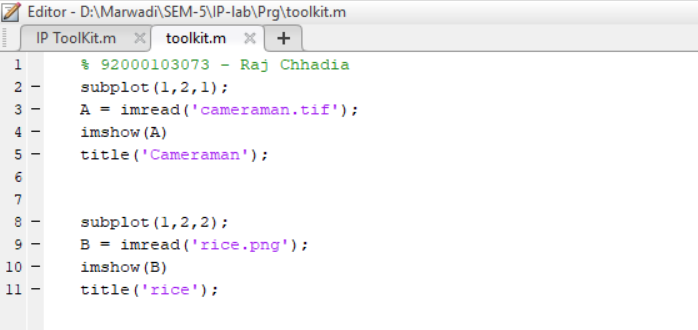
* + **size( ) command**
  + **imfinfo( ) command**

****

* + **impixel( ) command**

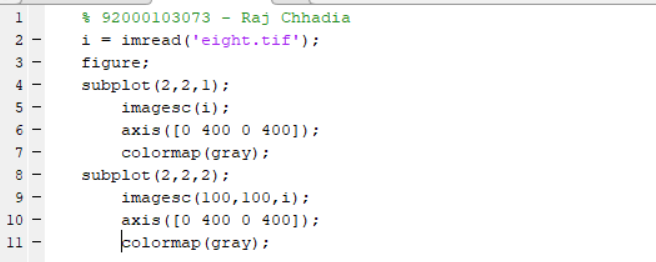
****

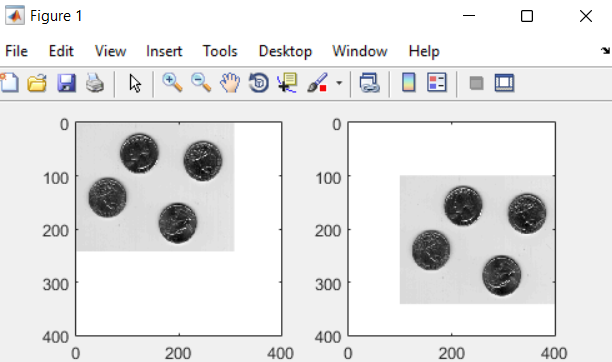
* + **subplot( ) command**



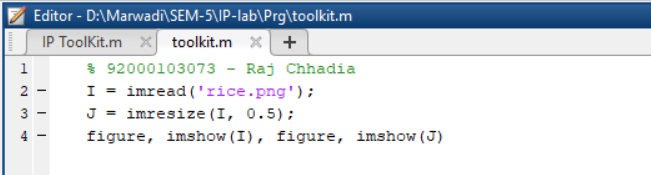
****

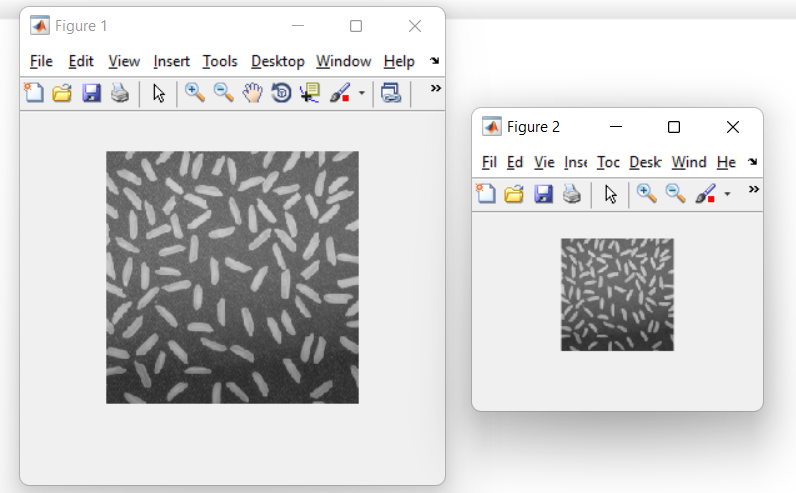
* + **imagesc( ) command**

****

****

* + **imresize( ) command**



****

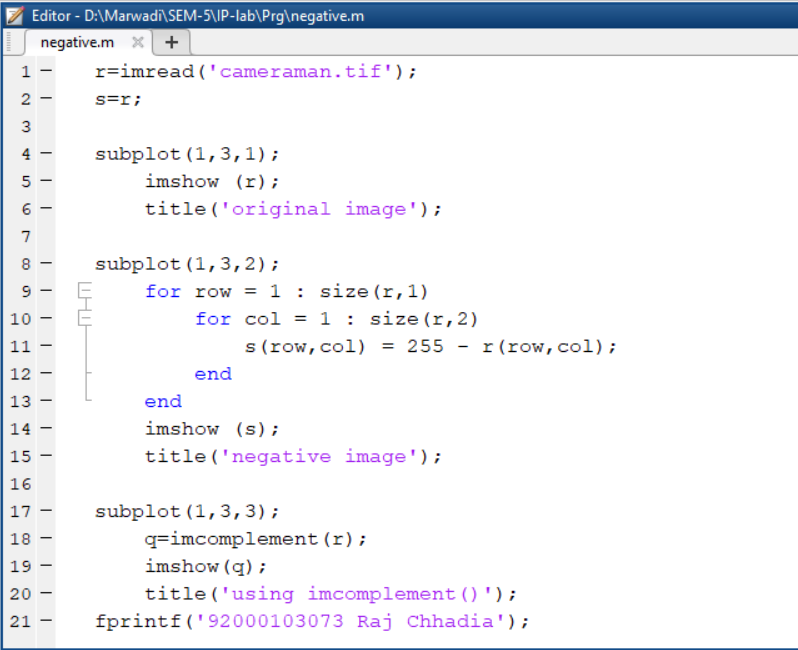
* + **imcrop( ) command**
  + **im2bw( ) command**
  + **rgb2gray( ) command**
  + **grayslice( ) command**
  + **imadd( ) command**
  + **imsubtract( ) command**
  + **imdivide( ) command**
  + **immultiply( ) command**
  + **imcomplement( ) command**
  + **uigetfile( ) command**

**Practical 2**

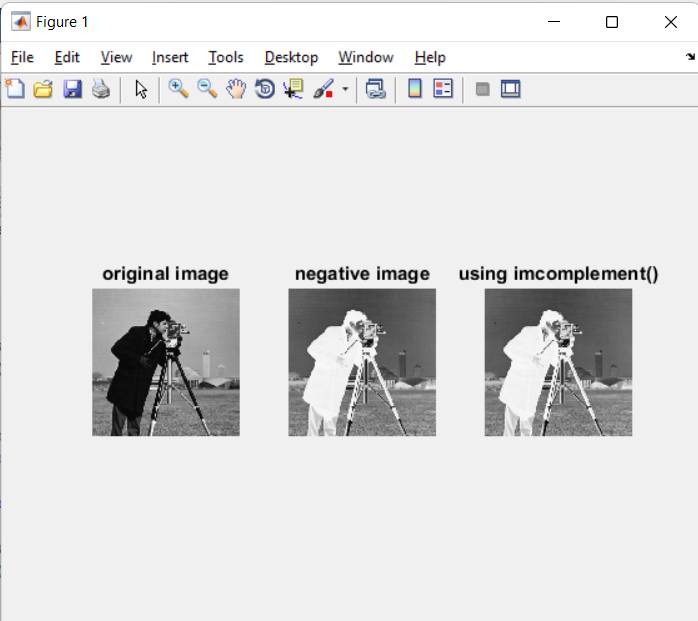
**Aim:** Point processing in spatial domain

1. Negation of an image
2. Thresholding of an image
3. Contrast Stretching of an image
4. **Negation of an image**

**Code:**

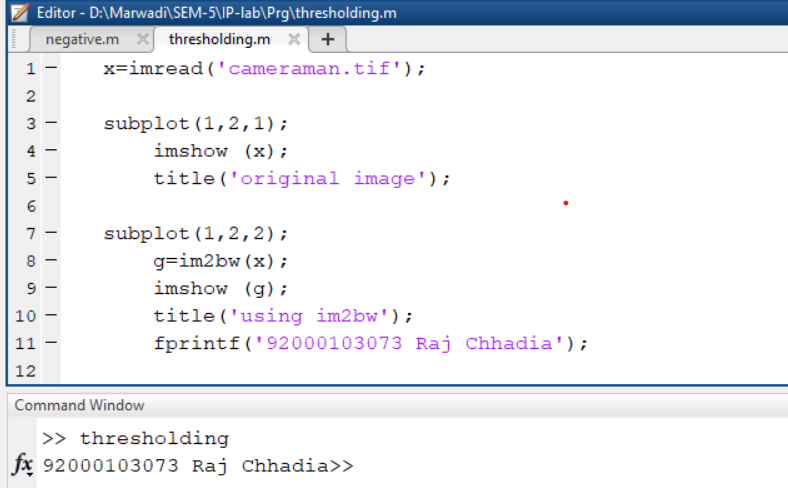
****

**Output:**

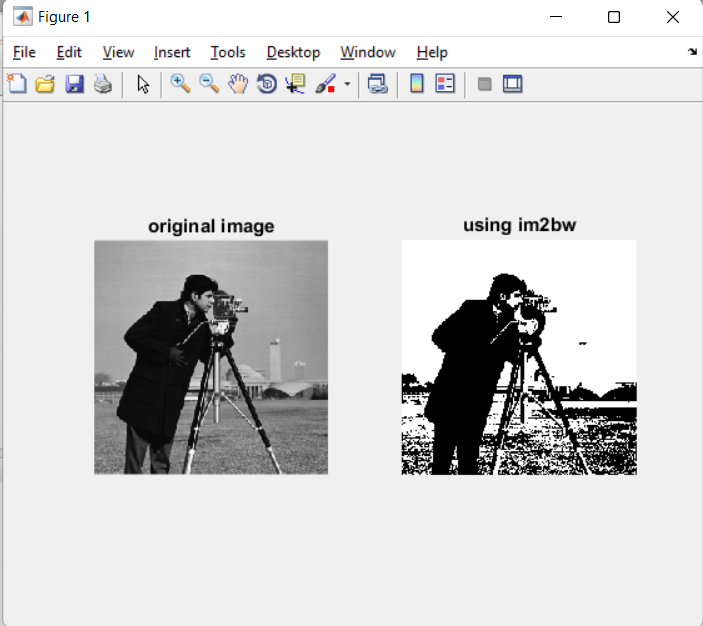


1. **Thresholding of an image**

**Code:**

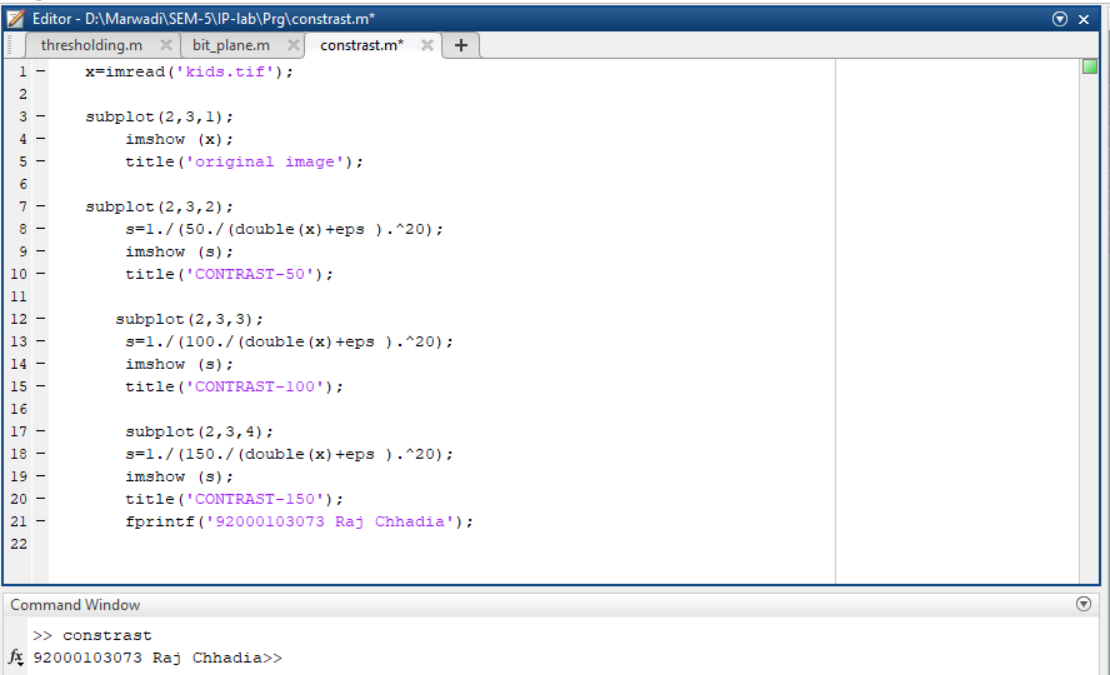
****

**Output:**

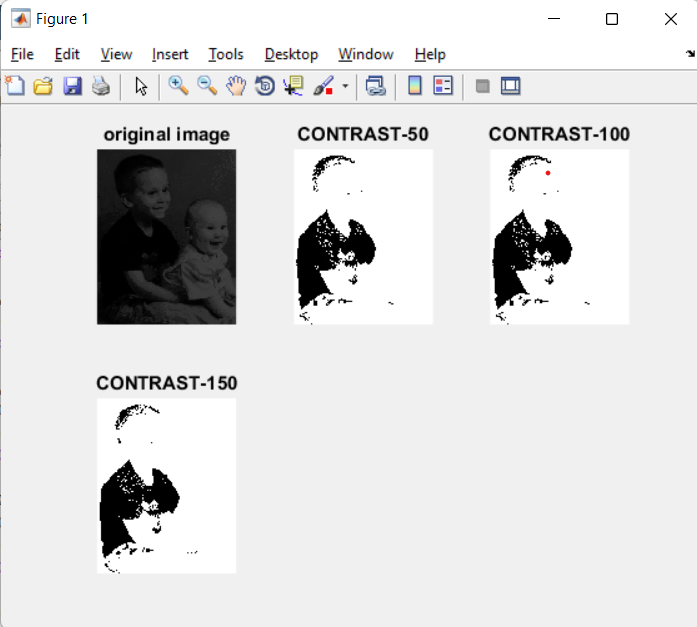


1. **Contrast Stretching of an image**

**Code:**



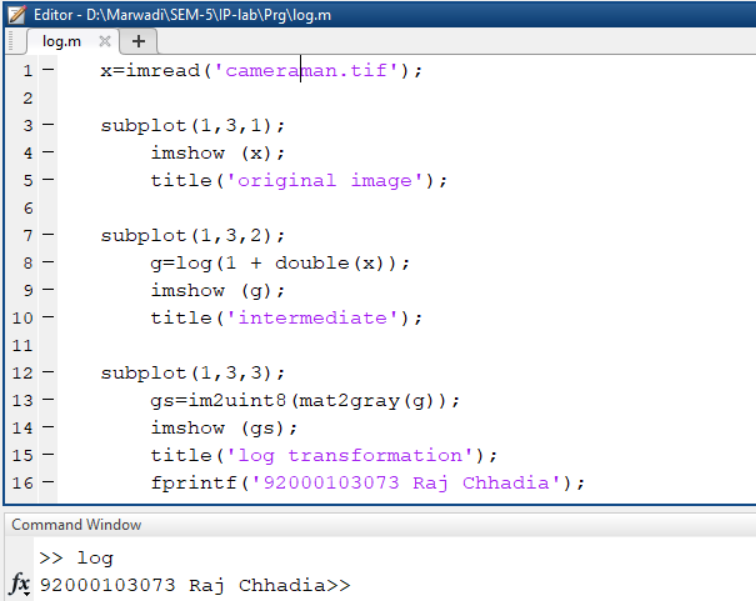
**Output:**

****

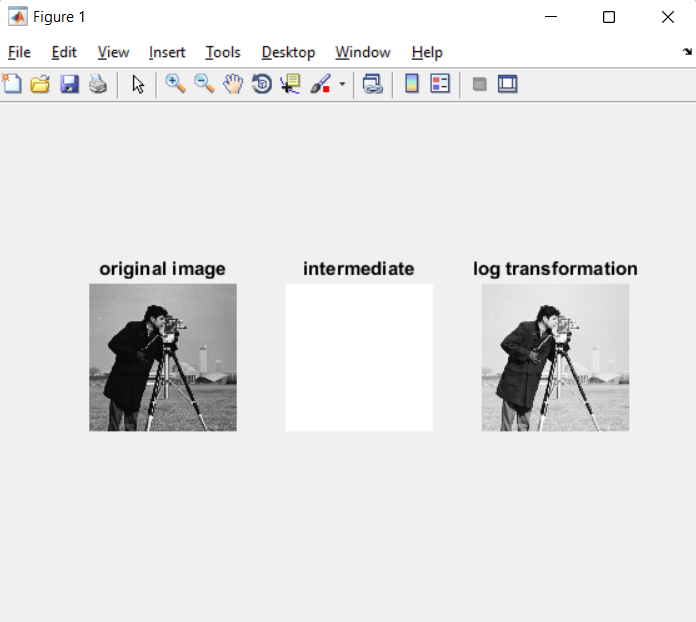
**Extra:**

1. Log Transformation
2. Power-Law Functions
3. Gray-level slicing
4. Bit-plane slicing
5. **Log Transformation**

**Code:**

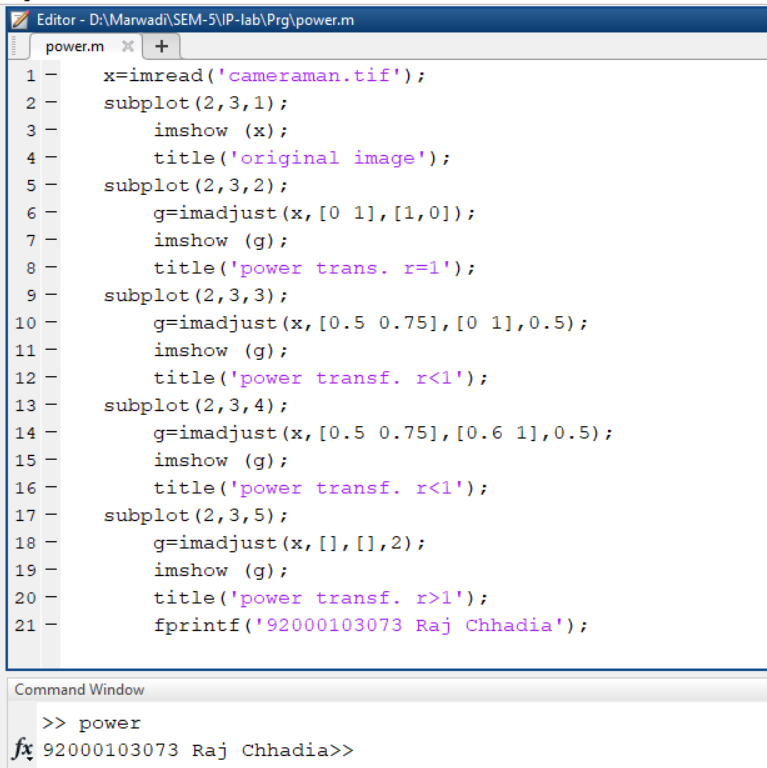
****

**Output:**

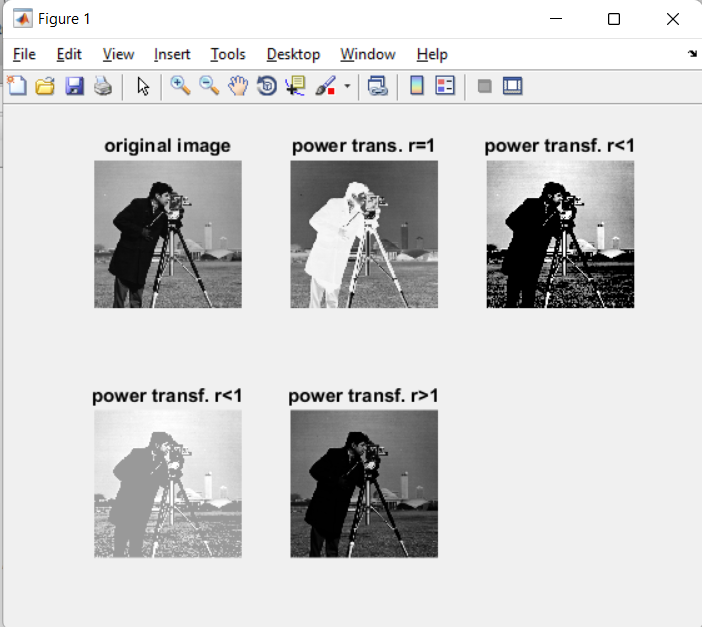


1. **Power-Law Functions**

**Code:**

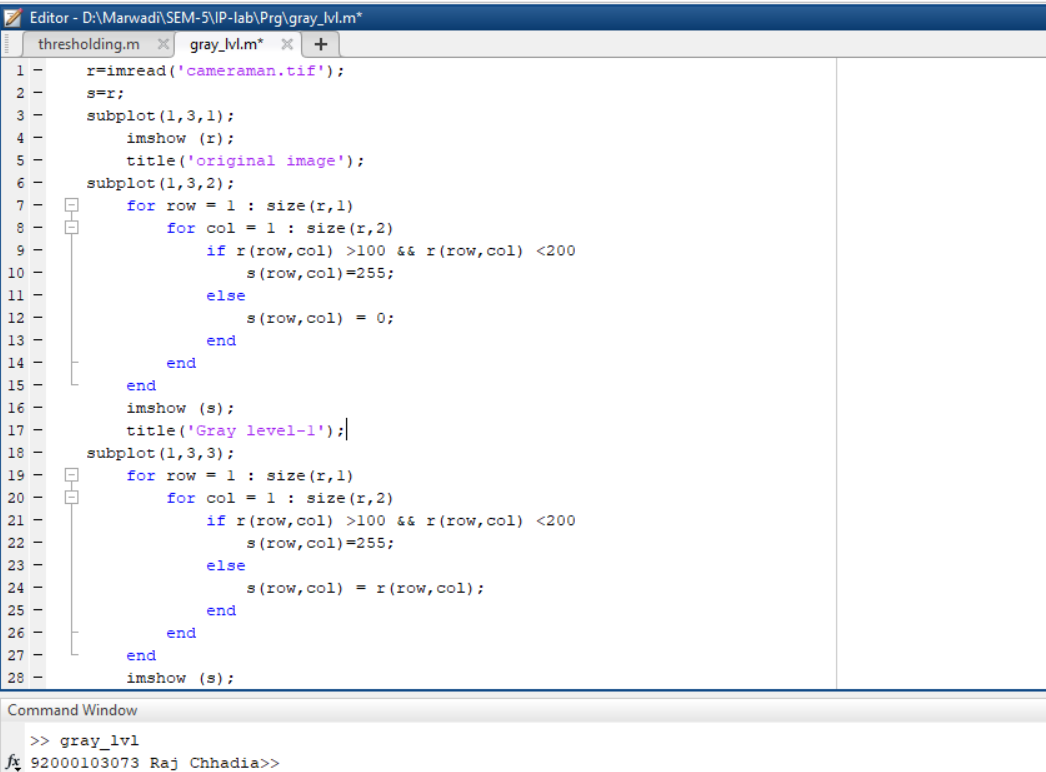
****

**Output:**

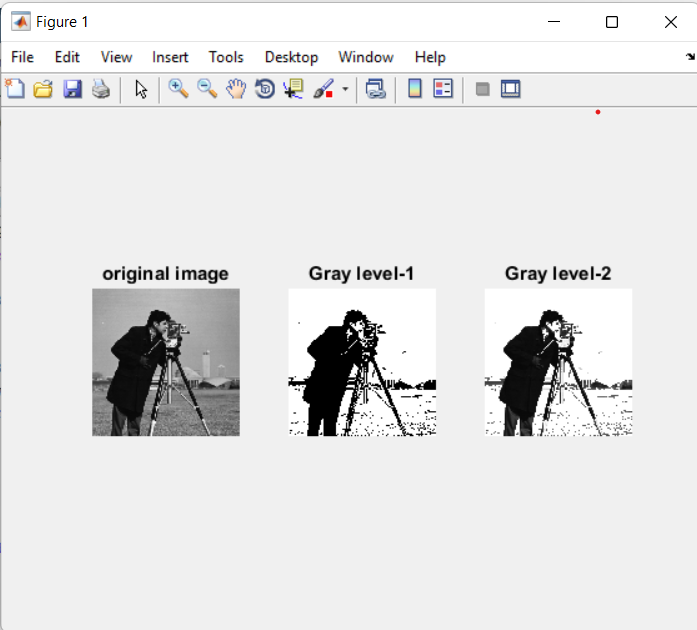


1. **Gray-level slicing**

**Code:**

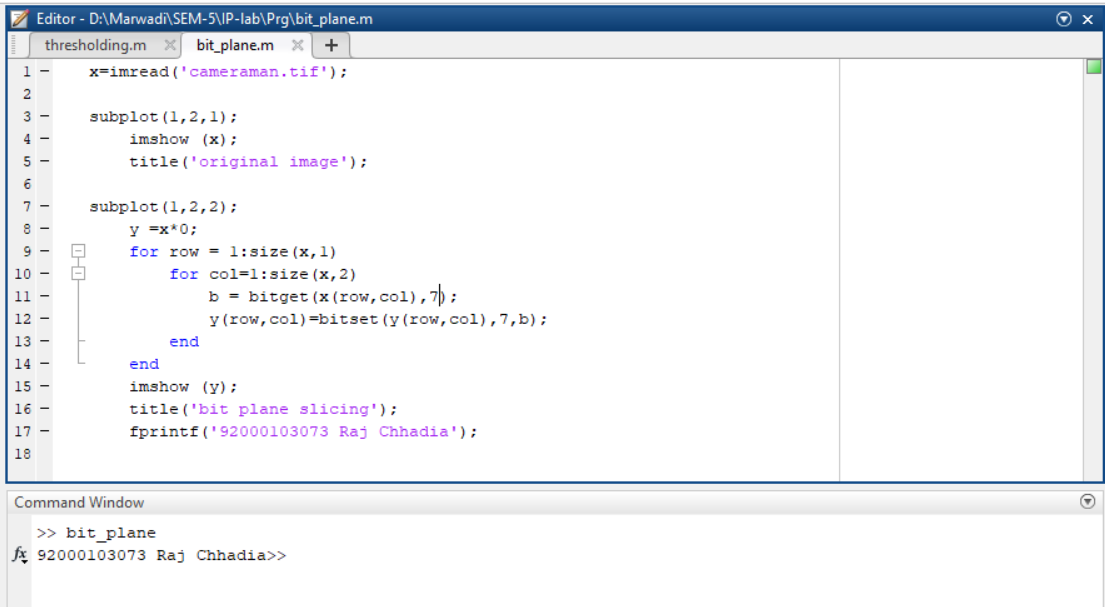


**Output:**

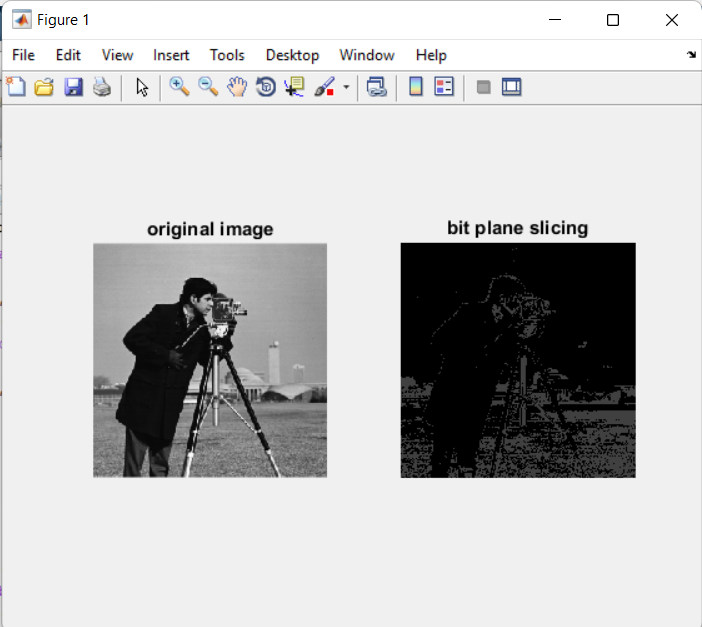


1. **Bit-plane slicing**

**Code:**



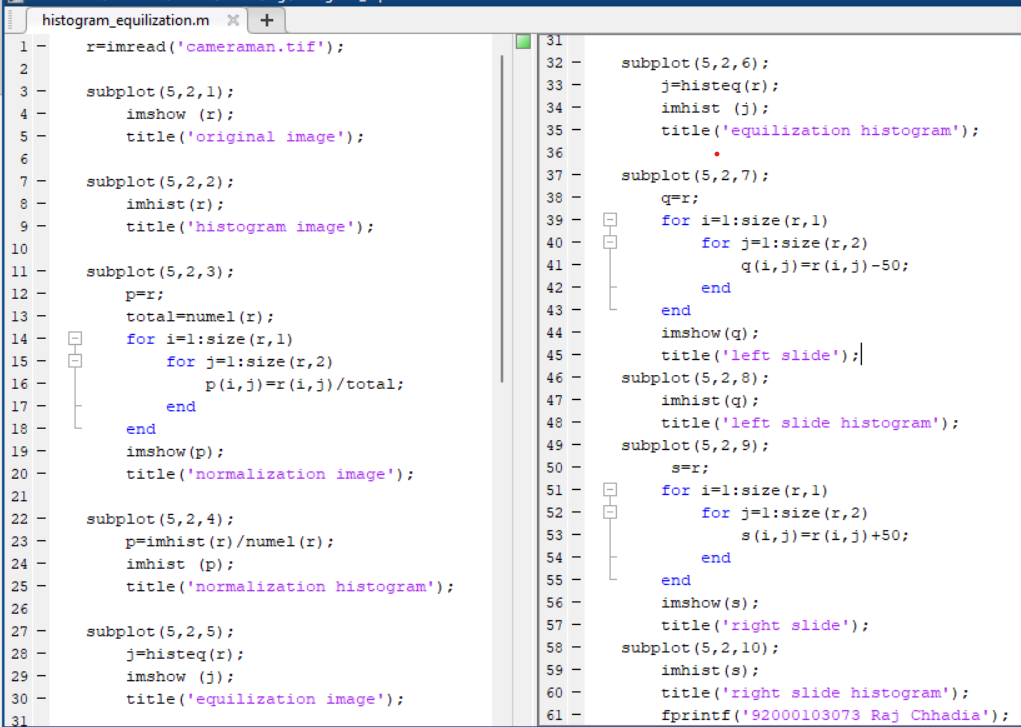
**Output:**



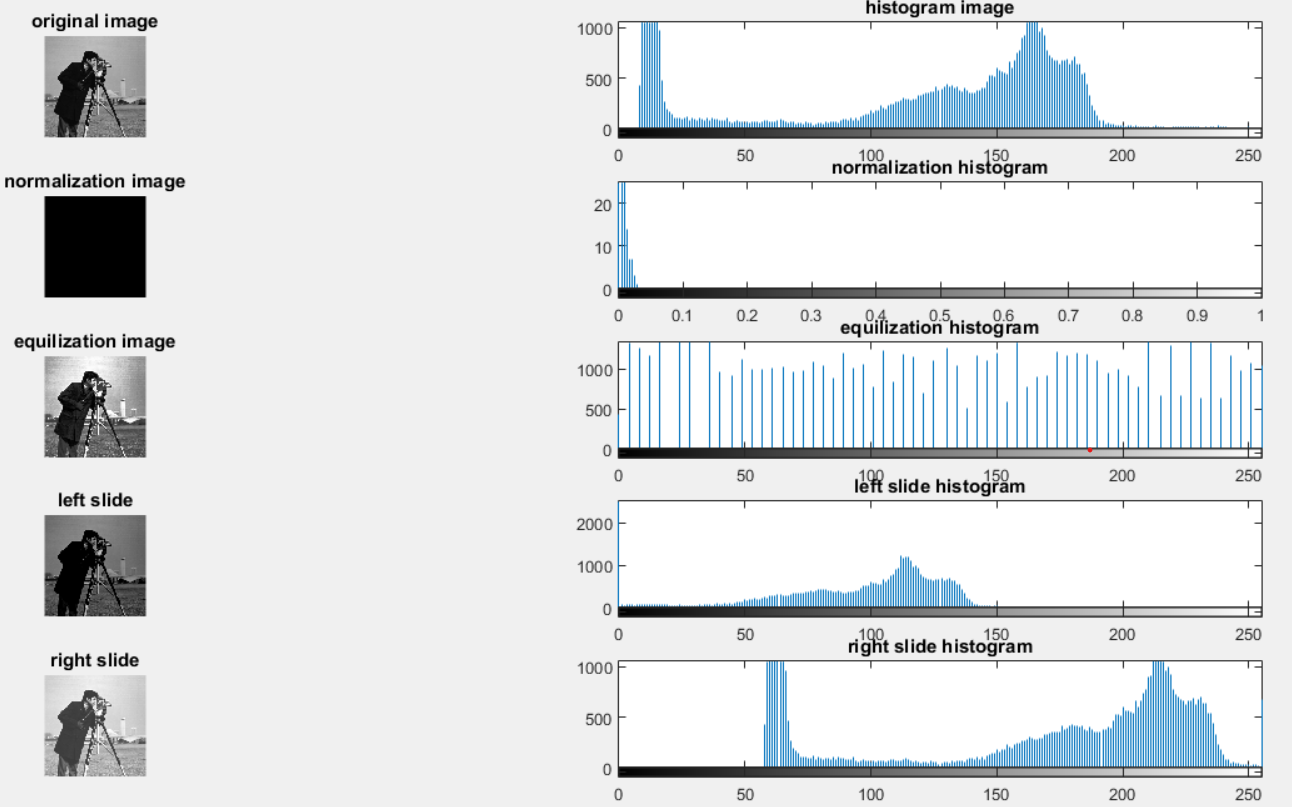
**Practical 3**

**Aim:** Write a program for histogram equalization.

**Code:**



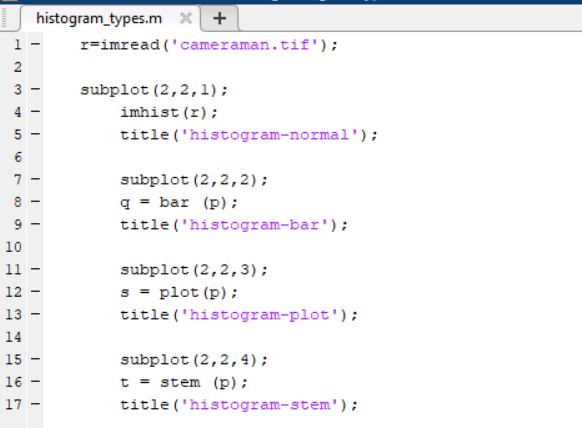
**Output:**



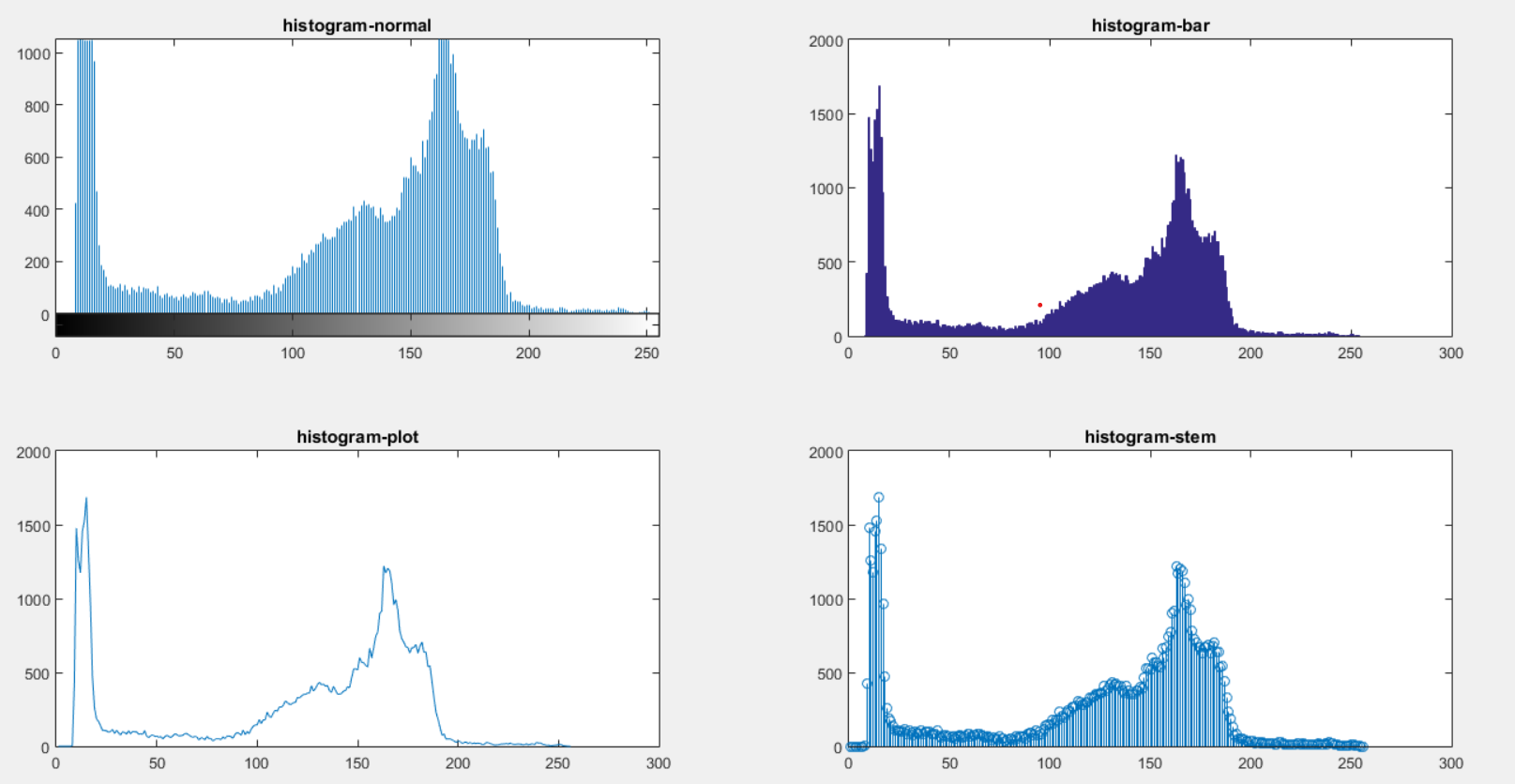
**Extra:**

1. **Histogram Types**

**Code:**

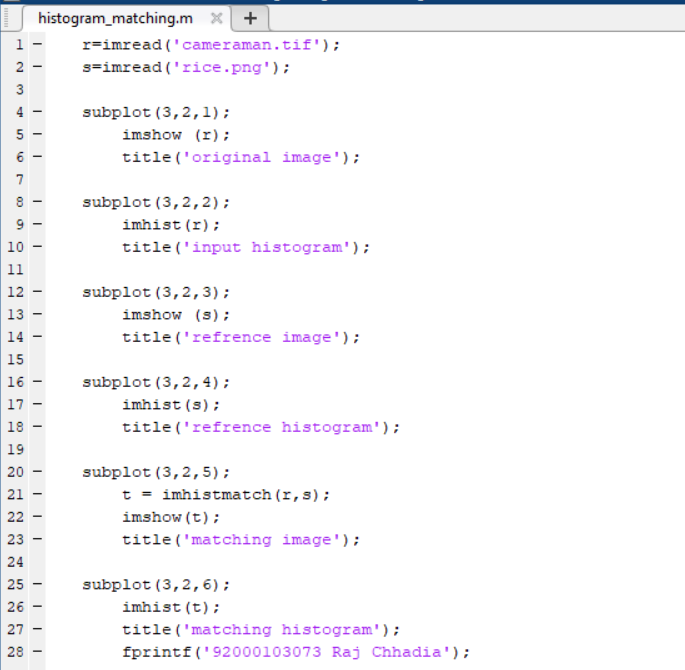


**Output:**



1. **Histogram Matching**

**Code:**



**Output:**

