

Output

```

>> A = [1,2,3,4;5,6]
>> A(:,2)
ans =
  2
  4
  6
>> A(1,:)
ans =
  1  2
>> A(1:2,:)
ans =
  1  2
  3  4
>> A(:,1:2)
ans =
  1  2
  3  4
  5  6
>> A(1:2,1:2)
ans =
  1
  3
>> length(A)
ans = 6
>> diag(A)
ans = [3 2]
>> reshape(A,1,6)
ans = [1 2 3 4 5 6]
>> diag(A)
ans = [1 4]

```

Expt. No. _____

Date _____

Page No. 1

Experiment No. - 1

Objective : To study and demonstrate MATLAB commands

colon used for a matrix

A(:,n) → refers to all elements in all rows of column n of matrix A

A(n,:) → refers to all columns in row n of matrix A

A(:,m:n) → refers to elements in all rows between columns m and n of matrix A

A(m:n,:) → refers to elements in all the columns between rows m and n of matrix A

A(min,p:n) → refers to elements in rows m through n and column p through q of the matrix A

Built-in functions for handling arrays

length(A) → return no. of elements in vector

size(A) → returns a row vector [m,n] where m = no. of rows, n = no. of columns

reshape(A,m,n) → rearrange a matrix A that has a rows and b columns to have m rows and n columns

r times s must be equal to m times n

diag(v) → creates a square matrix with elements between in diagonal

diag(A) → when A is a matrix, create a vector from the diagonal elements of A.

Teacher's Signature

General Purpose Commands

Operators and Special Characters

+	Plus; addition operator.
-	Minus; subtraction operator.
*	Scalar and matrix multiplication operator.
.*	Array multiplication operator.
^	Scalar and matrix exponentiation operator.
.^	Array exponentiation operator.
\	Left-division operator.
/	Right-division operator.
.\	Array left-division operator.
./	Array right-division operator.
:	Colon; generates regularly spaced elements and represents an entire row or column.
()	Parentheses; encloses function arguments and array indices; overrides precedence.
[]	Brackets; enclosures array elements.
.	Decimal point.
...	Ellipsis; line-continuation operator.
,	Comma; separates statements and elements in a row.
;	Semicolon; separates columns and suppresses display.
%	Percent sign; designates a comment and specifies formatting.
'	Quote sign and transpose operator.
.	Nonconjugated transpose operator.
=	Assignment (replacement) operator.

Commands for Managing a Session

clc	Clears Command window.
clear	Removes variables from memory.
exist	Checks for existence of file or variable.
global	Declares variables to be global.
help	Searches for a help topic.
lookfor	Searches help entries for a keyword.
quit	Stops MATLAB.
who	Lists current variables.
whos	Lists current variables (long display).

Special Variables and Constants

<code>ans</code>	Most recent answer.
<code>eps</code>	Accuracy of floating-point precision.
<code>i, j</code>	The imaginary unit $\sqrt{-1}$.
<code>Inf</code>	Infinity.
<code>NaN</code>	Undefined numerical result (not a number).
<code>pi</code>	The number π .

System and File Commands

<code>cd</code>	Changes current directory.
<code>date</code>	Displays current date.
<code>delete</code>	Deletes a file.
<code>diary</code>	Switches on/off diary file recording.
<code>dir</code>	Lists all files in current directory.
<code>load</code>	Loads workspace variables from a file.
<code>path</code>	Displays search path.
<code>pwd</code>	Displays current directory.
<code>save</code>	Saves workspace variables in a file.
<code>type</code>	Displays contents of a file.
<code>what</code>	Lists all MATLAB files in the current directory.
<code>wk1read</code>	Reads .wk1 spreadsheet file.

Input/Output and Formatting Commands**Input/Output Commands**

<code>disp</code>	Displays contents of an array or string.
<code>fscanf</code>	Read formatted data from a file.
<code>format</code>	Controls screen-display format.
<code>fprintf</code>	Performs formatted writes to screen or file.
<code>input</code>	Displays prompts and waits for input.
<code>;</code>	Suppresses screen printing.

Expt. No. _____

Output

```

>> a = [1, 2, 3, 4]
>> b = [3, 4, 5, 6, 7]
>> cat(a, b)
ans = [3 4 5 6 7]
>> length(a)
ans = 4
>> f8d(a)
ans = 2 3 4
>> linspace(-1, 1)
ans = column 1 through 7:
-1.0000 -0.9791 -0.9595 -0.9393 - - -
; ; ; ; ;
columns 99 and 100
0.9797 1.0000
>> logspace(-1, 1)
ans = column 1 through 8
0.1000 0.1099 0.1190 - - -
; ; ;
columns 49 and 50
3.1030 10.000
>> min(a)
ans = 1
>> prod(a)
ans = 24
>> sqrt(a)
ans = 1 4
>> sort(a)
ans = 1 2 3 4
>> sum(a)
ans = 10

```

Format Codes for fprintf and fscanf

%s	Format as a string.
%d	Format as an integer.
%f	Format as a floating point value.
%e	Format as a floating point value in scientific notation.
%g	Format in the most compact form: %f or %e.
\n	Insert a new line in the output string.
\t	Insert a tab in the output string.

Numeric Display Formats

format short	Four decimal digits (default).
format long	16 decimal digits.
format short e	Five digits plus exponent.
format long e	16 digits plus exponents.
format bank	Two decimal digits.
format +	Positive, negative, or zero.
format rat	Rational approximation.
format compact	Suppresses some line feeds.
format loose	Resets to less compact display mode.

Vector, Matrix and Array Commands**Array Commands**

cat	Concatenates arrays.
find	Finds indices of nonzero elements.
length	Computes number of elements.
linspace	Creates regularly spaced vector.
logspace	Creates logarithmically spaced vector.
max	Returns largest element.
min	Returns smallest element.
prod	Product of each column.
reshape	Change size.
size	Computes array size.
sort	Sorts each column.
sum	Sums each column.

Teacher's Signature _____

```

>> I=eye(3)    >> I=eye(2,3)    >> x= ones(2)    >> x=[1,4,2,5,3,6]
  1 0 0          1 0 0            x = 3x3           >> Sz= Sz(A)
  0 1 0          0 1 0            x = 1x2           Sz = 6.000 (A)
  0 0 1          1 1 1            x = 3x2           Sz = 1x2
                                         3 2
                                         1 1 1
                                         1 1 1    >> x = ones(3,2)

>> x=zeros(2)    >> A=[4,-2,1]
  x = 2x2          >> A = [4 -2 1]
  0 0              0 1
  0 0              1 1
  -2 -1 1 -2
  >> b = dot(A,x)
  D = 8

```

Expt. No. _____

Date _____
Page No. 5

Special Matrices

eye	Creates an identity matrix.
ones	Creates an array of ones.
zeros	Creates an array of zeros.

Matrix Arithmetic

cross	Computes cross products.
dot	Computes dot products.

Matrix Commands for Solving Linear Equations

det	Computes determinant of an array.
inv	Computes inverse of a matrix.
pinv	Computes pseudoinverse of a matrix.
rank	Computes rank of a matrix.
rref	Computes reduced row echelon form.

Cell Array Functions

cell	Creates cell array.
celldisp	Displays cell array.
cellplot	Displays graphical representation of cell array.
num2cell	Converts numeric array to cell array.
deal	Matches input and output lists.
iscell	Identifies cell array.

Structure Functions

fieldnames	Returns field names in a structure array.
getfield	Returns field contents of a structure array.
isfield	Identifies a structure array field.
isstruct	Identifies a structure array.
rmfield	Removes a field from a structure array.
setfield	Sets contents of field.
struct	Creates structure array.

Teacher's Signature _____

Expt. No. _____

Plot Enhancement Commands

<code>axes</code>	Creates axes objects.
<code>close</code>	Closes the current plot.
<code>close all</code>	Closes all plots.
<code>figure</code>	Opens a new figure window.
<code>gtext</code>	Enables label placement by mouse.
<code>hold</code>	Freezes current plot.
<code>legend</code>	Legend placement by mouse.
<code>refresh</code>	Redraws current figure window.
<code>set</code>	Specifies properties of objects such as axes.
<code>subplot</code>	Creates plots in subwindows.
<code>text</code>	Places string in figure.

Specialized Plot Commands

<code>bar</code>	Creates bar chart.
<code>loglog</code>	Creates log-log plot.
<code>polar</code>	Creates polar plot.
<code>semilogx</code>	Creates semilog plot (logarithmic abscissa).
<code>semilogy</code>	Creates semilog plot (logarithmic ordinate).
<code>stairs</code>	Creates stairs plot.
<code>stem</code>	Creates stem plot.

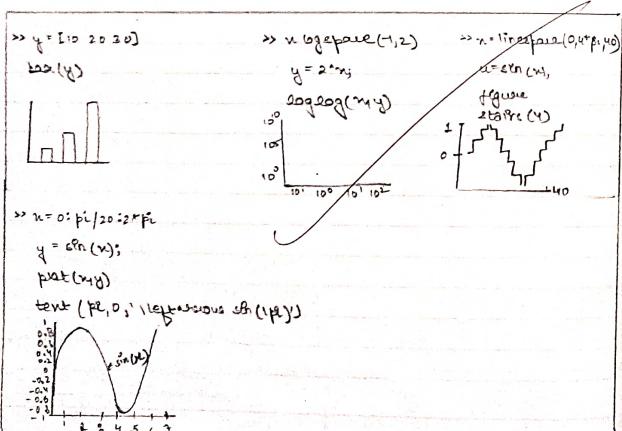
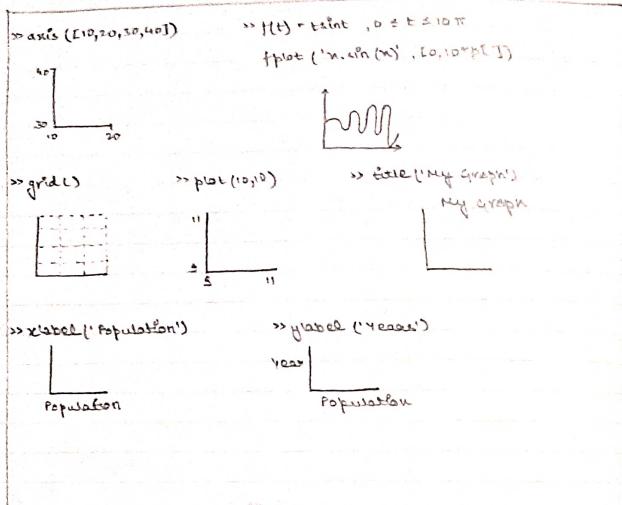
Histogram Functions

<code>bar</code>	Creates a bar chart.
<code>hist</code>	Aggregates the data into equally spaced bins.
<code>histc</code>	Aggregates the data into unequally spaced bins.

Programming**Logical and Relational Operators**

<code>==</code>	Relational operator: equal to.
<code>~=</code>	Relational operator: not equal to.
<code><</code>	Relational operator: less than.
<code><=</code>	Relational operator: less than or equal to.

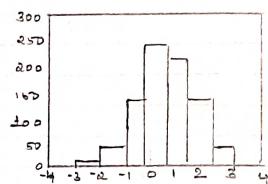
Teacher's Signature _____



Histogram function output

```
>> x = 2020;
y = [20 50 25];
b = bar(y, "stacked");
```

```
>> data = randn(1000, 1);
hist(data)
```



```
>> x = randn(100, 1);
b = range(-4:4;
[bincounts] = histc(x, b);
bincounts = 9x1
0
2
14
28
32
16
3
2
0
```

Date _____

Page No. 7.

Expt. No. _____

>	Relational operator: greater than.
>=	Relational operator: greater than or equal to.
&	Logical operator: AND.
	Logical operator: OR.
~	Logical operator: NOT.
xor	Logical operator: EXCLUSIVE OR.

Program Flow Control

break	Terminates execution of a loop.
case	Provides alternate execution paths within switch structure.
else	Delineates alternate block of statements.
elseif	Conditionally executes statements.
end	Terminates for, while, and if statements.
error	Display error messages.
for	Repeats statements a specific number of times.
if	Executes statements conditionally.
otherwise	Default part of switch statement.
return	Return to the invoking function.
switch	Directs program execution by comparing point with case expressions.
warning	Display a warning message.
while	Repeats statements an indefinite number of times.

Logical Functions

any	True if any elements are nonzero.
all	True if all elements are nonzero.
find	Finds indices of nonzero elements.
finite	True if elements are finite.
isnan	True if elements are undefined.
isinf	True if elements are infinite.
isempty	True if matrix is empty.
isreal	True if all elements are real.

M-Files

eval	Interpret strings containing Matlab expressions.
feval	Function evaluation.
function	Creates a user-defined function M-file.
global	Define global variables.
nargin	Number of function input arguments.
nargout	Number of function output arguments.

Teacher's Signature _____

Date _____

Expt. No. _____

Page No. ____ 8. _____

script**Script M-files**

Timing

cputime	CPU time in seconds.
clock	Current date and time as date vector.
tic, toc	Start, stop a stopwatch timer.

Mathematical Functions

Exponential and Logarithmic Functions

exp (x)	Exponential; e^x .
log (x)	Natural logarithm; $\ln(x)$.
log10 (x)	Common (base 10) logarithm; $\log(x) = \log_{10}(x)$.
sqrt (x)	Square root; \sqrt{x} .

Trigonometric Functions

acos (x)	Inverse cosine; $\arccos x = \cos^{-1}(x)$.
-----------------	--

Teacher's Signature _____

Output

```

>> v = [0.5 0 1 0.72]
erf(v)
ans = 0.5
ans = 0.8205 0 0.8427 0.6814

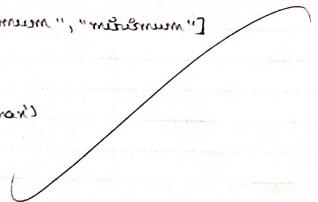
>> A = [0.11; 2.32; 3.01; ± 2.3]
>> mean(A)      >> median(A)      >> std(A)
ans =               ans =                  ans =
0.4667             1.5000              1.2910
2.3383             1.5000              1.2910
1.3383             1.5000              0.3574
2.0000

>> rand()
ans = 0.8147        >> randn()
ans = -0.7123

>> A = 3.4          >> fix(A)           >> floor(A)       >> round(A)
ceil(A)            ans = 3            ans = 2            ans = 3
ans = 4

>> strarray = ["max", "transliteration", "mintranslitr"]
>> findstr(strarray, "max")
ans = 1 2 11
>> strcmp('max', 'minmax')
ans = 1

```



Expt. No. _____

Complex Functions

abs(x)	Absolute value, $ x $.
angle(x)	Angle of a complex number x.
conj(x)	Complex conjugate of x.
imag(x)	Imaginary part of a complex number x.
real(x)	Real part of a complex number x.

Statistical Functions

erf(x)	Computes the error function $erf(x)$.
mean	Calculates the average.
median	Calculates the median.
std	Calculates the standard deviation.

Random Number Functions

rand	Generates uniformly distributed random numbers between 0 and 1.
randn	Generates normally distributed random numbers.

Numeric Functions

ceil	Rounds to the nearest integer toward +.
fix	Rounds to the nearest integer toward zero.
floor	Rounds to the nearest integer toward -.
round	Rounds towards the nearest integer.
sign	Signum function.

String Functions

findstr	Finds occurrences of a string.
strcmp	Compares strings.
char	Creates character string array

Teacher's Signature _____

Numerical Methods

Polynomial and Regression Functions

<code>conv</code>	Computes product of two polynomials
<code>deconv</code>	Computes ratio of polynomials.
<code>eig</code>	Computes the eigenvalues of a matrix.
<code>poly</code>	Computes polynomial from roots.
<code>polyfit</code>	Fits a polynomial to data.
<code>polyval</code>	Evaluates polynomial and generates error estimates.
<code>roots</code>	Computes polynomial roots.

Interpolation Functions

<code>interp1</code>	Linear and cubic-spline interpolations of a function of one variable.
<code>interp2</code>	Linear interpolation of a function of two variables.
<code>spline</code>	Cubic-spline interpolation.
<code>unmkpp</code>	Computes the coefficients of cubic-spline polynomials.

Root Finding and Minimization

<code>fmin</code>	Finds minimum of single-variable function.
<code>fmins</code>	Finds minimum of multivariable function.
<code>fzero</code>	Finds zero of single-variable function.

Numerical Integration Functions

<code>quad</code>	Numerical integration with adaptive Simpson's rule.
<code>quadl</code>	Numerical integration with adaptive Lobatto quadrature.
<code>trapz</code>	Numerical integration with the trapezoidal rule.

Numerical Differentiation Functions

<code>diff(x)</code>	Computes the difference between adjacent elements in the vector x.
<code>polyder</code>	Differentiates a polynomial, a polynomial product, or a polynomial quotient.

Teacher's Signature

Output

```

>> Enter marks : 75
      B
  
```

Q Find out the greatest among 4 using nested if else.

```

prompt='Enter a no : '
a=input(prompt)
prompt='Enter a no : '
b=input(prompt)
prompt='Enter a no : '
c=input(prompt)
prompt='Enter a no : '
d=input(prompt)
if a>b
    if a>c
        if a>d
            disp(a)
        else
            disp(d)
    else
        if c>d
            disp(c)
        else
            disp(d)
end.
  
```

Output

```

Enter a no : 2
Enter a no : 4
Enter a no : 3
Enter a no : 1
4
  
```

Expt. No. 2(a)

Date _____
Page No. 11

Experiment No-2(a)

Objective : Display the grade for the input marks

Syntax
 if expression
 statements
 elseif expression
 statements
 else
 statements
 end

Program
 prompt='Enter marks : '
 marks = input(prompt)
 if marks > 90
 disp('A+')
 elseif marks > 85
 disp('A')
 elseif marks > 80
 disp('B+')
 elseif marks > 75
 disp('B')
 elseif marks > 65
 disp('C+')
 elseif marks > 55
 disp('C')
 else
 disp('F')
 end

Teacher's Signature

Output

```
>> Enter a number: 5
Prime Number
```

Expt. No. 2(b)

Date _____
Page No. 12

Experiment No. 2(b)

Objective: Write a program using for and while loop to find out whether input number is prime or not.

Syntax for for loop

```
for index-variable  
statements  
end
```

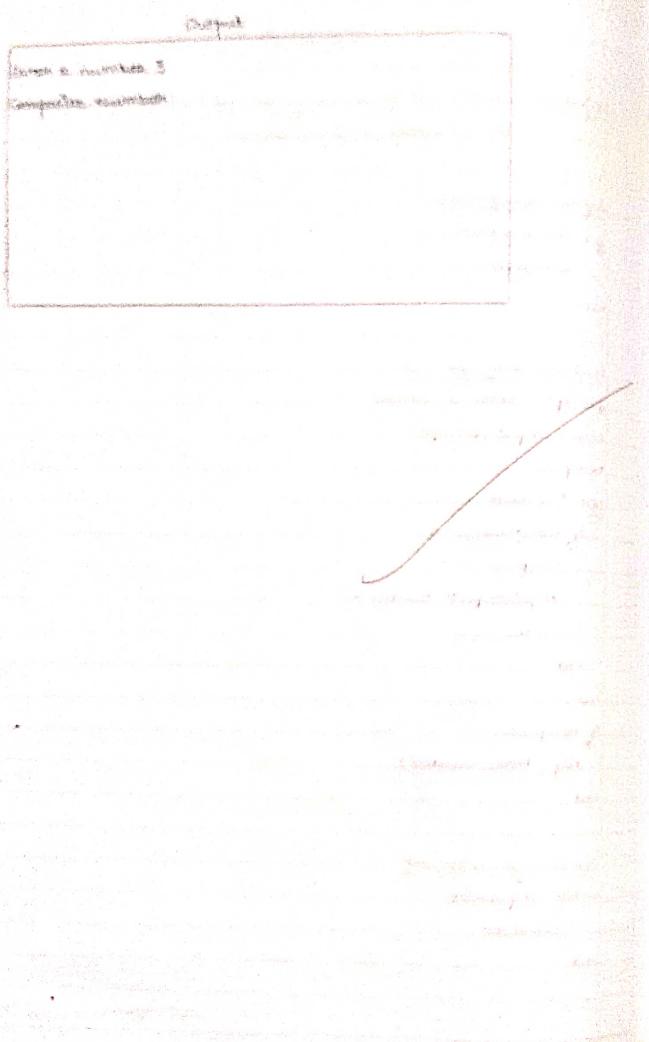
Program using for loop

```
prompt = 'Enter a number'  
num = input(prompt)  
temp = 0  
for i=2:num-1  
    if mod(num,i) == 0  
        temp = 1  
    disp('Composite Number!');  
    break;  
end  
end  
if temp == 0:  
    disp('Prime Number!')
```

Syntax for while loop

```
while expression  
statements  
end
```

Teacher's Signature _____



Date _____

Page No. _____ Page No. 13

```
Program using while loop
prompt = "Enter a number"
num = int(input(prompt))
count = 0
for i in range(1, num+1):
    print(i)
    count += 1
    if count == num:
        break
else:
    print("Loop completed")
print("End of program")
```

Teacher's Signature  172

Output

```

2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50

```

Expt. No. 21(3)

Date _____
Page No. m

Experiment: 20.3.2013

Objective: write a program to print even no. from 1 to 100 using
for loop

Program

```

for i=1:100
if mod(i,2)=0
    disp(i)
end
end

```

Teacher's Signature _____

Output

```
Enter input : A
uppercase letter
```

Date _____

Expt. No. 2(a)

Page No. 15

Experiment No. 2(a)

objective: write a program to have whatever input is digit, lowercase, uppercase or symbol using switch case

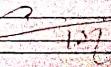
Syntax

```
switch switch expression
    case case expression
        statements
    case case expression
        statements
    ...
    otherwise
        statements
end
```

Program

```
prompt = ('Enter Input');
num = Read(prompt, 'c');
num = ord(num)
switch num
    case 49,48,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,
        117,118,119,120,121,122:
        disp('Lowercase letter');
    case 65,66,73,78,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,
        87,88,89,90:
        disp('Uppercase letter');
    case 48,49,50,51,52,53,54,55,56,57:
        disp('Digit');
    otherwise:
        disp('Symbol');
end
```

Teacher's Signature _____



Expt. No. 3

Experiment No.: 3

Objective: To study & implement Image commands using MATLAB

1. imread: Read image from graphic file
 >> I = imread('image.jpg');

2. imshow: Display image
 >> imshow(I)

3. im2bw: Set variables in workspace, with -size and type
 >> im2bw(I)

4. imfinfo: Information about graphic file
 >> imfinfo('image.jpg')

5. imwrite: Write image to graphic file
 >> imwrite(A,'image.jpg')

Program

```
I = imread('fort.jpg')
imshow(I)
im2bw(I)
imwrite(I,'fort.jpg')
imfinfo('fort.jpg')
```

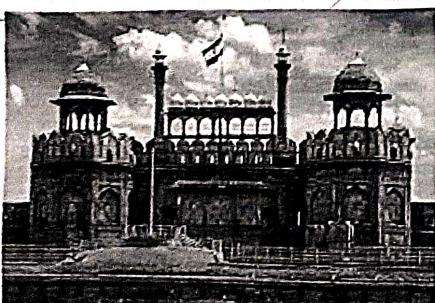
6. imager: Scale data and display image object

7. improfile: Pixel value, cross-sections along line segments

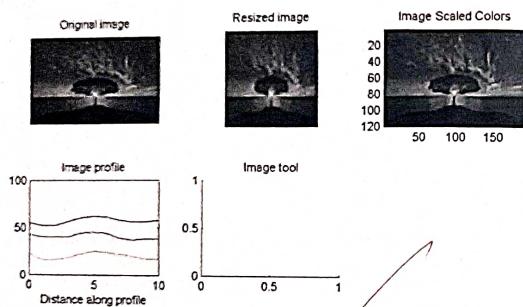
Teacher's Signature _____

Output

```
>> who('I')
Name    Size         Class      Attributes
I      1x1x600x3  uint8-B  [  ] % [  ] % [  ] %
>> imfinfo('fort.jpg')
File name : 'C:\Downloads\fort.jpg'
File date : '16-Mar-2023 16:19:24'
File size : 67048
Format : 'jpg'
Format version : ''
  width : 600
  height : 600
  bit depth : 24
Color Type : 'Truecolor'
Format signature : ''
Number of samples : 3
Coding Method : 'Huffman'
Coding Process : 'Sequential'
Comments : #3
```



Expt. No. _____



c. imtool: Open a new image tool

a. imresizr: Resize Image

Program

```
n = imread ('C:\Users\student\landscape.jpg');
subplot (3,3,1)
imshow (n); title ('original Image')
```

subplot (3,3,2)

```
imageres (n); title ('image1')
```

subplot (3,3,3)

```
imtool (n); title ('imtool')
```

subplot (3,3,4)

```
improfile (n, [20 30], [30,40]); title ('improfile')
```

subplot (3,3,5)

```
y = imresizr (n, 0.5);
imshow (y); title ('imresizr')
```

To Insert image through dialog box

```
filename = uigetfile ('*.jpg; *.tif; *.png; *.gif'; 'All Image files';
'*.*; 'All files', 'Dialog Box!');

A = imread (filename);
imshow (A);
```

Teacher's Signature _____

Expt. No. 3

Experiment No. 4Objective : To study and implement the usage of function

1. write a program to find the factorial of a number

```
function n = fact(n)
if n<=1
    n=1;
else
    n=n*fact(n-1);
end
end
```

2. write a program to find greatest among five numbers

```
function a=greatest(num,num1,num2,num3,num4)
if ((num>num1) && (num>num2) && (num>num3) && (num>num4))
    a=num
elseif ((num1>num) && (num1>num2) && (num1>num3) && (num1>num4))
    a=num1
elseif ((num2>num) && (num2>num1) && (num2>num3) && (num2>num4))
    a=num2
elseif ((num3>num) && (num3>num1) && (num3>num2) && (num3>num4))
    a=num3
else
    a=num4
end
end
```

Teacher's Signature

Output

```
>> fact(5)
ans =
120
```

Output

```
>> greatest(4,6,8,2,5)
ans =
8
```

Expt. No. _____

Teacher's Signature

Q. Write a program to find mean, median and variance of given numbers.

```

function [avg, med, var] = mystats(x)
n = length(x);
x = sort(x);
avg = nummean(x);
function avg = nummean(x)
sum = 0;
for i = 1:n
    sum = sum + x(i);
end
avg = sum/n;
end
med = mymedian(x);
function med = mymedian(x)
if mod(n, 2) == 0
    med = (x(n/2) + x(n/2 + 1))/2;
else
    med = x(n/2);
end
end
var = myvar(x);
function var = myvar(x)
tmp = mymean(x);
res = 0;
for i = 1:n
    res = res + (x(i) - tmp) * (x(i) - tmp);
end
var = res/(n-1);
end

```

Output:

>> Z = [10 20 30 40]

```

>> mystats(Z)
ans =
    25
    25
    166.6667
    25

```

Expt. No. _____

Output

»fibonacci(20)

ans =

0 1 2 3 5 8 13

```
dBip(aug);
dPip(mod);
dPip(vas);
end
```

4. create a function to print fibonacci series from 1 to 20

function fib = fibonacci(n)

fib = [0 1];

while fib(end) + fib(end-1) <= n

fib(end+1) = fib(end) + fib(end-1);

end

end

Teacher's Signature



Objectives : To convert image in multiple forms

Program

```

I = imread ('C:\Users\student\Downloads\image1.jpg');
subplot(3,3,1);
imshow(I); title('Original Image');

subplot(3,3,2);
J = rgb2gray(I);
imshow(J); title('Gray Image');

subplot(3,3,3);
K = Bwblue(I);
imshow(K); title('B&W Image');

subplot(3,3,4);
L = rgb2hsv(I);
imshow(L); title('HSV Image');

subplot(3,3,5);
T_red = I;
T_red(:,:,2)=0;
T_red(:,:,3)=0;
imshow(T_red); title('Red Image');

subplot(3,3,6);
T_green = I;
T_green(:,:,1)=0;
T_green(:,:,3)=0;
imshow(T_green); title('Green Image');

subplot(3,3,7);

```

Teacher's Signature _____

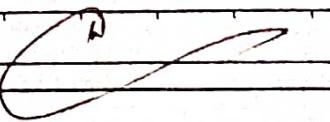
Date _____

Expt. No. _____

Page No. 22

 $I_{blue} = I;$ $I_{blue}(i, j, 1) = 0;$ $I_{blue}(i, j, 2) = 0;$ $\text{imshow}(I_{blue}); \text{title}(\text{'Blue Image'}) ;$

Teacher's Signature



Experiment No: 6

Objective : To study and implement the basic arithmetic operations on images.

Program

```

subplot(3,3,1);
I = imread('moon2.jpg');
x = imresize(I,[100,200]);
imshow(x); title('Image 1');

subplot(3,3,2);
J = imread('moony.jpg');
y = imresize(J,[100,200]);
imshow(y); title('Image 2');

subplot(3,3,3);
add = imadd(x,y);
imshow(add); title('Addition');

subplot(3,3,4);
sub = imsubtract(x,y);
imshow(sub); title('Subtraction');

subplot(3,3,5);
mul = immultiply(x,y);
imshow(mul); title('Multiplication');

subplot(3,3,6);
div = imdivide(x,y);
imshow(div); title('Division');

subplot(3,3,7);
com = imcomplement(x);
imshow(com); title('Complement')

```

Experiment No. -Objective :Program

```

[x, map] = imread ('Index.png');
J = ind2gray (x, map);
subplot (3,3,1);
imshow (x, map); title ("Indexed Image");
subplot (3,3,2);
imshow (J); title ("Grayscale Image");
K = imread ('grayrose.jpg');
rgb = ind2rgb (x, map);
subplot (3,3,3);
imshow (rgb); title ('RGB Image');
[L, map] = rgb2ind (rgb, 300);
subplot (3,3,4);
imshow (L); title ("Indexed");
colormap (map);
T = im2hsv (rgb);
subplot (3,3,5);
imshow (T); title ("Black & White");
U = imread ('matrix.jpg');
subplot (3,3,6);
imshow (U); title ("Matrix Image");
V = filter2 ('fspecial (''sobel''), U);
S = mat2gray (V);
subplot (3,3,7);
imshow (S); title ('Gray Image');

```

Teacher's Signature _____

Experiment - BObjective :Program

```
imagedata = imread('grayrose.jpg');
subplot(3,3,1);
imshow(imagedata); title('gray Image');
[rows,columns] = size(imagedata);
for i=1:rows
    for j=1:columns
        if (imagedata(i,j) < 100)
            imagedata(i,j)=0;
        end
    end
end
subplot(3,3,2);
imshow(imagedata); title('Enhance Image');
```