Digital Signal Processing Laboratory

EXPERIMENT – 01 (MATLAB Basics)

Shorya Sharma
19EE01017

EXPERIMENT – 01 (MATLAB Basics)

Question 1: Basic Matrix Operations

a) Create a simple vector with 10 elements called A.
   a. MATLAB Program:

```
A=rand(10,1)
```

   b. Output:

```
A =

    0.1576
    0.9706
    0.9572
    0.4854
    0.8003
    0.1419
    0.4218
    0.9157
    0.7922
    0.9595
```

b) Add 3 to each element of vector A and store the result in a new vector B.
   a. MATLAB Program:

```
B = A+3
```

   b. Output:

```
B =

    3.1576
    3.9706
    3.9572
    3.4854
    3.8003
    3.1419
    3.4218
    3.9157
    3.7922
    3.9595
```

c) Creating a 5 x 5 matrix 'X' using semicolons (;) to separate the rows of a matrix.

a. MATLAB Program:

```
X = [1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15; 16 17 18 19 20; 21 22 23 24 25]
```

b. Output:

```
X =

     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
    16    17    18    19    20
    21    22    23    24    25
```

d) Find the transpose of matrix 'X' and store the result in a new matrix 'Y'.

a. MATLAB Program:

```
Y = X'
```

b. Output:

```
Y =

     1     6    11    16    21
     2     7    12    17    22
     3     8    13    18    23
     4     9    14    19    24
     5    10    15    20    25
```

e) Multiply these two matrices, 'X' and 'Y'.

a. MATLAB Program:

```
X * Y
```

b. Output:

```
ans =

    55   130   205   280   355
   130   330   530   730   930
   205   530   855  1180  1505
   280   730  1180  1630  2080
   355   930  1505  2080  2655
```

f) Multiply the corresponding elements of the two matrices.

a. MATLAB Program:

```
X .* Y
```

b. Output:

```
ans =

     1    12    33    64   105
    12    49    96   153   220
    33    96   169   252   345
    64   153   252   361   480
   105   220   345   480   625
```

g) Create a column vector 'X1', from the elements of the 4th row of matrix 'X'.

a. MATLAB Program:

```
X1 = (X(4,:))' % transpose of the 4th row of X assigned to X1
```

b. Output:

```
X1 =

    16
    17
    18
    19
    20
```

h) Select the elements in mth through nth columns of matrix 'X'.

a. MATLAB Program:

```
X(:,2:5) % here m=2, n=5
```

b. Output:

```
ans =

     2     3     4     5
     7     8     9    10
    12    13    14    15
    17    18    19    20
    22    23    24    25
```

i) Create a sub matrix 'X2' taking the inner subpart of matrix 'X'.

a. MATLAB Program:

```
X(3,:) = [] % deleting the third row
```

b. Output:

```
X =

     1     2     3     4     5
     6     7     8     9    10
    16    17    18    19    20
    21    22    23    24    25
```

Question 2: MATLAB Commands: Graph, Save and Read

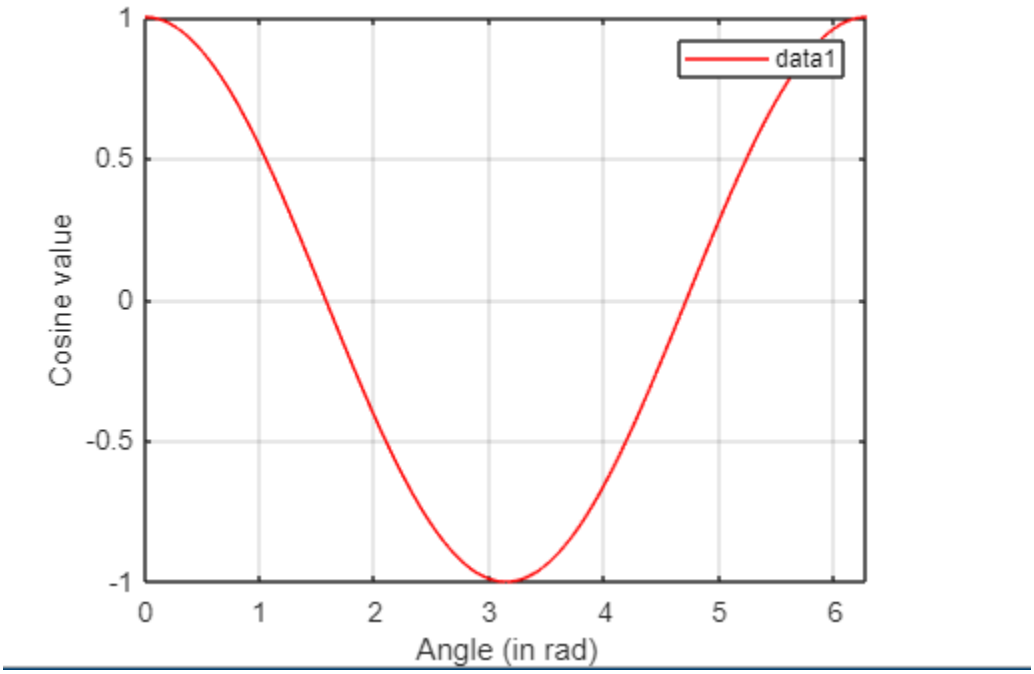
a) Generate elementary signals and display using ***plot***, ***subplot*** and ***stem*** commands in MATLAB. Also add ***xlabel***, ***ylabel***, ***axis tight***, ***grid on***, ***legend*** to the figure.

a. Using ***plot*** command:

i. MATLAB Program:

```
x = 0:pi/100:2*pi;
y = cos(x);
plot(x,y,'-r')
xlabel('Angle (in rad)')
ylabel('Cosine value')
grid on
axis tight
legend
```

ii. Output/Figure:

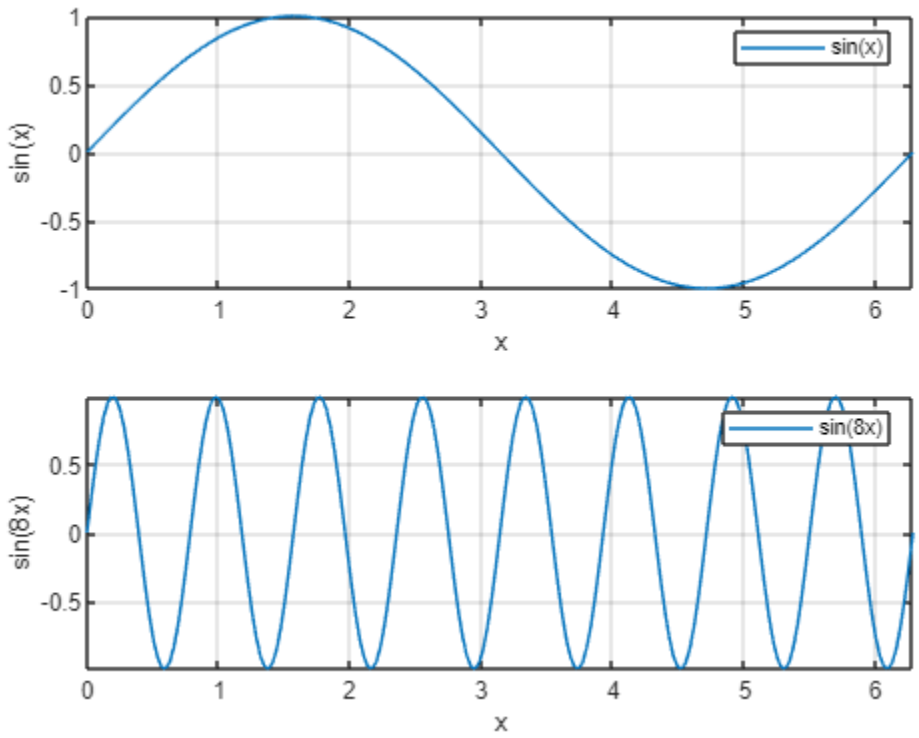


b. Using ***subplot*** command:

i. MATLAB Program:

```
subplot(2,1,1)
x = 0:pi/100:2*pi;
y1 = sin(x);
plot(x,y1)
xlabel('x')
ylabel('sin(x)')
grid on
axis tight
legend('sin(x)')


subplot(2,1,2)
y4 = sin(8*x);
plot(x,y4)
xlabel('x')
ylabel('sin(8x)')
grid on
axis tight
legend('sin(8x)')
```

ii. Output/Figure:

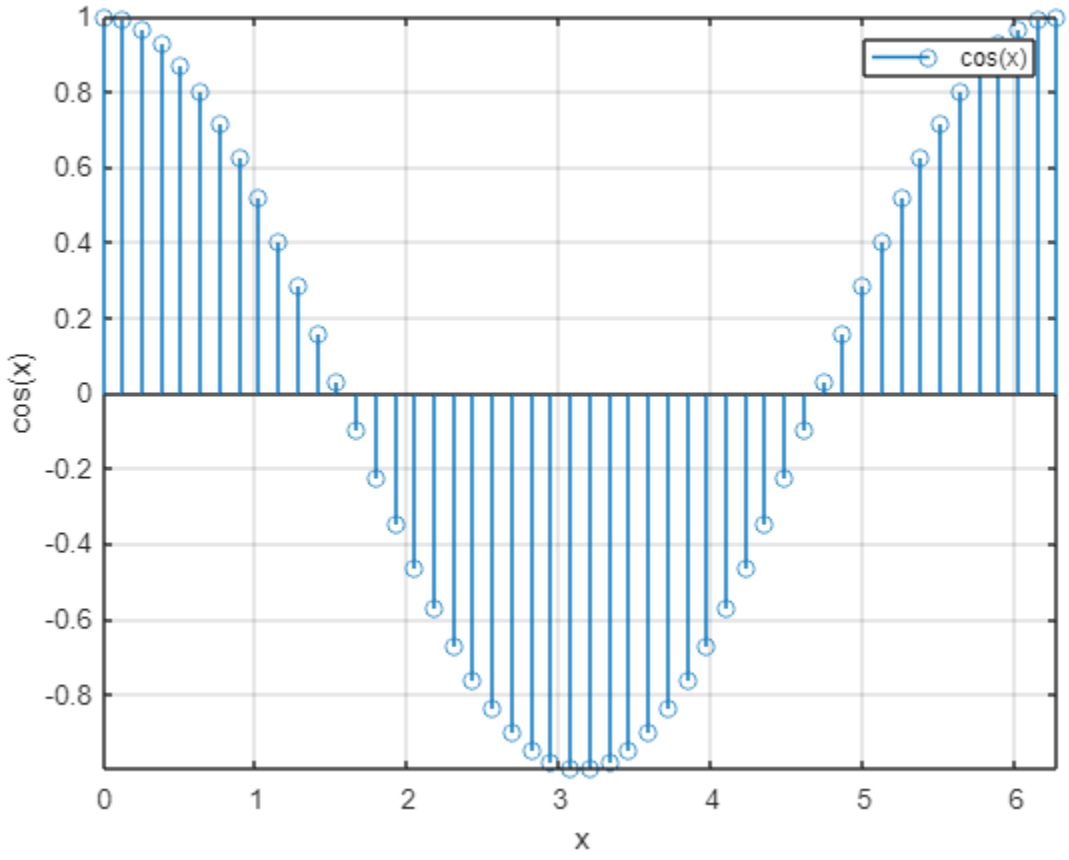


c. Using ***stem*** command:

i. MATLAB Program:

```
X = linspace(0,2*pi,50)';
Y = cos(X);
stem(X,Y)
xlabel('x');
ylabel('cos(x)');
grid on
axis tight
legend('cos(x)')
```

ii. Output/Figure:



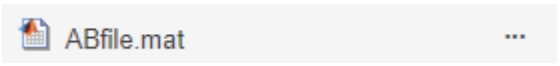
b) ***save, load, xlsread, xlswrite, audioread, audiowrite*** commands:

a. Create and save two variables, A and B, to a file ABfile.mat:

i. MATLAB Program:

```
A = rand(5,1);
B = ones(2,3);
save('ABfile.mat','A','B')
```

ii. Output:

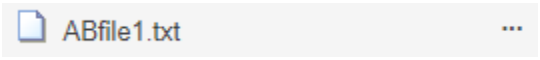


c. Create and save two variables A and B to a file ABfile1.txt:

i. MATLAB Program:

```
A = rand(5,1);
B = ones(2,3);
save('ABfile1.txt','A','B')
```

ii. Output:

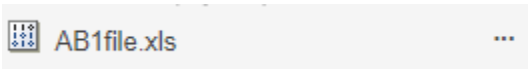


c. Create and save two variables A and B to a file AB1file.xls:

i. MATLAB Program:

```
A = rand(5,1);
B = ones(2,3);
save('AB1file.xls','A','B')
```

ii. Output:



d. Create two variables, save them to an ASCII file, and then view the contents of the file.

i. MATLAB Program:

```
A = rand(5,1);
B = ones(2,3);
save('ascii_file.txt','A','B', '-ascii')
type('ascii_file.txt')
```

ii. Output:

```
7.5774013e-01
7.4313247e-01
3.9222702e-01
6.5547789e-01
1.7118669e-01
1.0000000e+00   1.0000000e+00   1.0000000e+00
1.0000000e+00   1.0000000e+00   1.0000000e+00
```

e. Create a structure s1 that contains three fields a, b, c. Save the fields of structure s1 as indivisual variables in a file called newstruct.mat.

i. MATLAB Program:

```
s1.a = 4;
s1.b = {'abc',[4 5; 6 7]};
s1.c = 'variable c';
save('newstruct.mat','-struct','s1');
disp('Contents of newstruct.mat:')
whos('-file','newstruct.mat')
```

ii. Output:

```
Contents of newstruct.mat:
  Name      Size            Bytes  Class     Attributes

  a         1x1                 8  double
  b         1x2               246  cell
  c         1x10               20  char
```

f. Save two variables to example.mat and then append the third variable.

i. MATLAB Program:

```
p = rand(1,10);
q = ones(5);
save('example.mat','p','q')
a = 50;
save('example.mat','a','-append')
whos('-file','example.mat')
```

ii. Output:

```
Name      Size            Bytes  Class     Attributes

a         1x1                 8  double
p         1x10               80  double
q         5x5               200  double
```

c) ***readcell*** and ***writecell*** commands.

a. ***writecell:***

i. MATLAB Program:

```
C =  {1,2,3;
     'text',datetime('today'),hours(1)}
 writecell(C)
type 'C.txt'
```

ii. Output:

```
C =

  2×3 cell array

    {[   1]}    {[          2]}    {[   3]}
    {'text'}    {[22-Aug-2020]}    {[1 hr]}


1,2,3
text,22-Aug-2020,1 hr
```

b. ***readcell:***

i. MATLAB Program:

```
C = readcell('C.txt')
```

ii. Output:

```
C =

  2×3 cell array

    {[   1]}    {[          2]}    {[   3]}
    {'text'}    {[22-Aug-2020]}    {'1 hr'}
```

d) *writematrix, readmatrix* commands:

a. ***writematrix:***

i. MATLAB Program:

```
M = rand(5,5);
writematrix(M,'M.txt','Delimiter','tab')
type M.txt
```

ii. Output:

```
0.957506835434298   0.485375648722841   0.792207329559554   0.933993247757551   0.655477890177557
0.964888535199277   0.8002804688888 0.959492426392903   0.678735154857773   0.171186687811562
0.157613081677548   0.141886338627215   0.655740699156587   0.757740130578333   0.706046088019609
0.970592781760616   0.421761282626275   0.0357116785741896  0.743132468124916   0.0318328463774207
0.957166948242946   0.915735525189067   0.849129305868777   0.392227019534168   0.27692298496089
```

b. ***readmatrix:***

i. MATLAB Program:

```
A = readmatrix('M.txt')
```

ii. Output:

```
A =

    0.9575    0.4854    0.7922    0.9340    0.6555
    0.9649    0.8003    0.9595    0.6787    0.1712
    0.1576    0.1419    0.6557    0.7577    0.7060
    0.9706    0.4218    0.0357    0.7431    0.0318
    0.9572    0.9157    0.8491    0.3922    0.2769
```

e) Read and write to audio files in a range of different formats like .wav, .mp4 etc.

a. MATLAB Program:

```
load audio1.mat
audiowrite('audio1.wav',y,Fs);
audiowrite('audio2.mp4',y,Fs);
clear y Fs

%read data back into matlab
[y,Fs]=audioread('audio1.wav');
[y,Fs]=audioread('audio2.mp4');
%listen to the audio
sound(y,Fs);
```

Question 3: Writing Expressions

a) Express the quadratic equation with "a", "b", and "c" and its quadratic formula in MATLAB. Also find the roots of the quadratic equation by assuming the weights "a", "b" and "c" of the expression.

a. MATLAB Program:

```
syms a b c x;
equation = a*x^2+b*x+c == 0
solve(equation)

%Assuming weights a, b, c;
a = 3;
b = 2;
c = 1;
eqn = a*x^2+b*x+c == 0
S = solve(eqn,x)
```

b. Output:

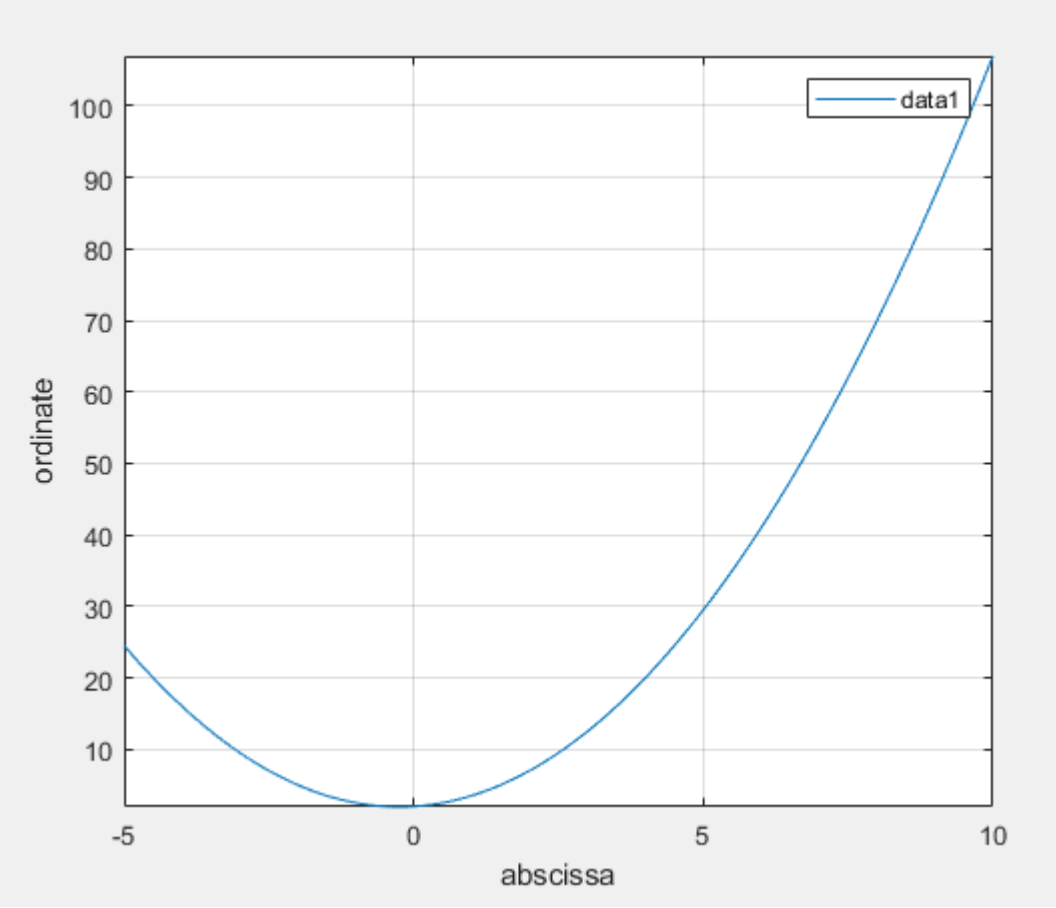
```
equation =

a*x^2 + b*x + c == 0


ans =

 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)


eqn =

3*x^2 + 2*x + 1 == 0


S =

 - (2^(1/2)*1i)/3 - 1/3
   (2^(1/2)*1i)/3 - 1/3
```

b) Plot the function $y = x^2 + 0.5x + 2$ over the range x = [-5, 10].

a. MATLAB program:

```
x = -5:0.01:10;
y = x.^2+0.5*x+2;
plot(x,y)
xlabel('abscissa')
ylabel('ordinate');
grid on
axis tight
legend
```

b. Output:



Question 4: MATLAB Commands: Create, Combine Arrays, Manipulations

a) repelem – Repeat copies of array elements.
   a. MATLAB Program:

```
v = rand(2,1)
u = repelem(v,3)
```

   b. Output:

```
v =

    0.1576
    0.9706


u =

    0.1576
    0.1576
    0.1576
    0.9706
    0.9706
    0.9706
```

b) repmat – Repeat copies of array.
   a. MATLAB Program:

```
A = magic(2)
B = repmat(A, 2, 3)
```

   b. Output:

```
A =

     1     3
     4     2


B =

     1     3     1     3     1     3
     4     2     4     2     4     2
     1     3     1     3     1     3
     4     2     4     2     4     2
```

c) eye – Create identity matrix.
   a. MATLAB Program:

```
A = eye(4)
```

   b. Output:

```
A =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

d) zeros – Create array of all zeros.
   a. MATLAB Program:

```
A = zeros(4,5)
```

   b. Output:

```
A =

     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
```

e) ones – Create array of all ones.
   a. MATLAB Program:

```
A = ones(4,5)
```

   b. Output:

```
A =

     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

f) rand – Uniformly distributed random numbers.

a. MATLAB Program:

```
A = rand(4,5)
```

b. Output:

```
A =

    0.9572    0.4218    0.6557    0.6787    0.6555
    0.4854    0.9157    0.0357    0.7577    0.1712
    0.8003    0.7922    0.8491    0.7431    0.7060
    0.1419    0.9595    0.9340    0.3922    0.0318
```

g) reshape – Reshape array.

a. MATLAB Program:

```
A = 1:8;
a = reshape(A,[4,2])
B = magic(2);
b = reshape(B,[],1)
C = zeros(3,2,3);
c = reshape(C,2,9)
```

b. Output:

```
a =

     1     5
     2     6
     3     7
     4     8


b =

     1
     4
     3
     2


c =

     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0
```

h) cat – Concatenate arrays.

a. MATLAB Program:

```
A = ones(2);
B = zeros(2);
cat(1,A,B)
cat(2,A,B)
```

b. Output:

```
ans =

     1     1
     1     1
     0     0
     0     0


ans =

     1     1     0     0
     1     1     0     0
```

i) horzcat – Concatenate arrays horizontally.

a. MATLAB Program:

```
A = ones(2)
B = zeros(2,3)
horzcat(A,B)
```

b. Output:

```
A =

     1     1
     1     1


B =

     0     0     0
     0     0     0


ans =

     1     1     0     0     0
     1     1     0     0     0
```

j) vercat – Concatenate arrays vertically.

a. MATLAB Programs:

```
A = ones(3)
B = zeros(2,3)
vertcat(A,B)
```

b. Output:

```
A =

     1     1     1
     1     1     1
     1     1     1


B =

     0     0     0
     0     0     0


ans =

     1     1     1
     1     1     1
     1     1     1
     0     0     0
     0     0     0
```

k) diag – Create diagonal matrix or get diagonal elements of a matrix.

a. MATLAB Program:

```
A = diag([1,5,4])
```

b. Output:

```
A =

     1     0     0
     0     5     0
     0     0     4
```

l) blkdiag – Block diagonal matrix.

a. MATLAB Program:

```
A1 = ones(1,2);
A2 = 2*ones(3,2);
A3 = 3*ones(3,3);
B = blkdiag(A1,A2,A3)
```

b. Output:

```
B =

     1     1     0     0     0     0     0
     0     0     2     2     0     0     0
     0     0     2     2     0     0     0
     0     0     2     2     0     0     0
     0     0     0     0     3     3     3
     0     0     0     0     3     3     3
     0     0     0     0     3     3     3
```

m) fliplr – Flips a matrix left to right.

a. MATLAB Program:

```
A1 = 1:5
fliplr(A1)
```

b. Output:

```
A1 =

     1     2     3     4     5


ans =

     5     4     3     2     1
```

n) flipud – Flips a matrix up and down.

a. MATLAB Program:

```
A1 = rand(4,3)
flipud(A1)
```

b. Output:

```
A1 =

    0.2769    0.6948    0.4387
    0.0462    0.3171    0.3816
    0.0971    0.9502    0.7655
    0.8235    0.0344    0.7952


ans =

    0.8235    0.0344    0.7952
    0.0971    0.9502    0.7655
    0.0462    0.3171    0.3816
    0.2769    0.6948    0.4387
```

o) rot90 – Rotate array counterclockwise.

a. MATLAB Program:

```
A1 = rand(4,3)
B = rot90(A1)
C = rot90(B)
```

b. Output:

```
A1 =

    0.1869    0.7094    0.6551
    0.4898    0.7547    0.1626
    0.4456    0.2760    0.1190
    0.6463    0.6797    0.4984


B =

    0.6551    0.1626    0.1190    0.4984
    0.7094    0.7547    0.2760    0.6797
    0.1869    0.4898    0.4456    0.6463


C =

    0.4984    0.6797    0.6463
    0.1190    0.2760    0.4456
    0.1626    0.7547    0.4898
    0.6551    0.7094    0.1869
```

p) tril – Extracts lower triangular part.

a. MATLAB Program:

```
A1 = rand(4,3)
tril(A1)
```

b. Output:

```
A1 =

    0.1493    0.8143    0.1966
    0.2575    0.2435    0.2511
    0.8407    0.9293    0.6160
    0.2543    0.3500    0.4733


ans =

    0.1493         0         0
    0.2575    0.2435         0
    0.8407    0.9293    0.6160
    0.2543    0.3500    0.4733
```

q) triu – Extracts upper triangular part.

a. MATLAB Program:

```
A1 = rand(4,3)
triu(A1)
```

b. Output:

```
A1 =

    0.3517    0.9172    0.3804
    0.8308    0.2858    0.5678
    0.5853    0.7572    0.0759
    0.5497    0.7537    0.0540


ans =

    0.3517    0.9172    0.3804
         0    0.2858    0.5678
         0         0    0.0759
         0         0         0
```

r) permute – Permute array dimensions.

a. MATLAB Program:

```
A1 = rand(1,3,2)
B = permute(A1,[2,3,1])
size(B)
```

b. Output:

```
A1(:,:,1) =

    0.8530    0.6221    0.3510


A1(:,:,2) =

    0.5132    0.4018    0.0760


B =

    0.8530    0.5132
    0.6221    0.4018
    0.3510    0.0760


ans =

     3     2
```

s) circshift – Shift array circularly.

a. MATLAB Program:

```
A = ones(2,2);
B = zeros(2,2);
X = cat(2,A,B)
Y = circshift(X,1,2)
```

b. Output:

```
X =

     1     1     0     0
     1     1     0     0


Y =

     0     1     1     0
     0     1     1     0
```

t) length – Length of the largest array dimension.

a. MATLAB Program:

```
A = rand(3,8)
length(A)
```

b. Output:

```
A =

    0.2399    0.2400    0.9027    0.4893    0.3692    0.3897    0.0965    0.9561
    0.1233    0.4173    0.9448    0.3377    0.1112    0.2417    0.1320    0.5752
    0.1839    0.0497    0.4909    0.9001    0.7803    0.4039    0.9421    0.0598


ans =

     8
```

u) size – Array size.

a. MATLAB Program:

```
A = rand(3,8)
size(A)
```

b. Output:

```
A =

    0.2348    0.0154    0.6491    0.4509    0.7447    0.1835    0.7802    0.7757
    0.3532    0.0430    0.7317    0.5470    0.1890    0.3685    0.0811    0.4868
    0.8212    0.1690    0.6477    0.2963    0.6868    0.6256    0.9294    0.4359


ans =

     3     8
```

v) ndims – Number of array dimensions.

a. MATLAB Program:

```
A = rand(3,8,2)
ndims(A)
```

b. Output:

```
A(:,:,1) =

    0.4468    0.5108    0.6443    0.5328    0.8759    0.5870    0.4709    0.1948
    0.3063    0.8176    0.3786    0.3507    0.5502    0.2077    0.2305    0.2259
    0.5085    0.7948    0.8116    0.9390    0.6225    0.3012    0.8443    0.1707


A(:,:,2) =

    0.2277    0.9234    0.9049    0.1111    0.5949    0.7112    0.2967    0.5079
    0.4357    0.4302    0.9797    0.2581    0.2622    0.2217    0.3188    0.0855
    0.3111    0.1848    0.4389    0.4087    0.6028    0.1174    0.4242    0.2625


ans =

     3
```

w) numel – Number of array elements.

a. MATLAB Program:

```
A = ones(3,5)
numel(A)
```

b. Output:

```
A =

     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1


ans =

    15
```

x) isempty – Determine whether array is empty.

a. MATLAB Program:

```
A = ones(3,5)
isempty(A)
B = zeros(0,0)
isempty(B)
```

b. Output:

```
A =

     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1


ans =

  logical

   0


B =

     []


ans =

  logical

   1
```

Question 5: MATLAB – Loop Types

a) while loop – repeatedly executes statements while condition is true.

a. MATLAB Program:

```
A = rand(3,5)
i=1;
j=1;
sum=0;
% sum of all elements of A using while
while i<4
    j=1;
    while j<6
        sum = sum + A(i,j);
        j = j+1;
    end
    i=i+1;
end
answer=sum
```

b. Output:

```
A =

    0.2760    0.1626    0.9597    0.2238    0.5060
    0.6797    0.1190    0.3404    0.7513    0.6991
    0.6551    0.4984    0.5853    0.2551    0.8909


answer =

    7.6023
```

b) for loop – a loop that needs to execute a number of times.

a. MATLAB Program:

```
s = 10;
H = zeros(s);
% Creating hilbert matrix of order 10
for c = 1:s
    for r = 1:s
        H(r,c) = 1/(r+c-1);
    end
end
H
```

b. Output:

```
H =

    1.0000    0.5000    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000
    0.5000    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909
    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833
    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769
    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714
    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667
    0.1429    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667    0.0625
    0.1250    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667    0.0625    0.0588
    0.1111    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667    0.0625    0.0588    0.0556
    0.1000    0.0909    0.0833    0.0769    0.0714    0.0667    0.0625    0.0588    0.0556    0.0526
```

c) Nested loops – for or/and while.

a. MATLAB Program:

```
A = rand(3,6)
sum=0;
% Computing sum of all elements of A
for i=1:3
    j=1;
    for j=1:6
        sum = sum + A(i,j);
    end
end
sum
```

b. Output:

```
A =

    0.9593    0.1493    0.2543    0.9293    0.2511    0.3517
    0.5472    0.2575    0.8143    0.3500    0.6160    0.8308
    0.1386    0.8407    0.2435    0.1966    0.4733    0.5853


sum =

    8.7888
```

d) continue – loop to skip the remainder of its body.

a. MATLAB Program:

```
for n = 1:20
% Displaying numbers between 1 and 20
% that are divisible by 3
    if mod(n,3)
        continue
    end
    disp(['Divisible by 3: ' num2str(n)])
end
```

b. Output:

```
Divisible by 3: 3
Divisible by 3: 6
Divisible by 3: 9
Divisible by 3: 12
Divisible by 3: 15
Divisible by 3: 18
```

e) break – terminates the loop statement.

a. MATLAB Program:

```
for n = 1:20
% Displaying numbers between 1 and 20
% that are divisible by 3 and less than 14
    if mod(n,3)
        continue
    end
    if n>=14
        break
    end
    disp(['Divisible by 3: ' num2str(n)])
end
```

b. Output:

```
Divisible by 3: 3
Divisible by 3: 6
Divisible by 3: 9
Divisible by 3: 12
```

Question 5: MATLAB – Decision Making

a) if … end statement.
   a. MATLAB Program:

```
n = 5
if n>4
    j=10;
end
j
```

   b. Output:

```
n =

     5


j =

    10
```

b) if … else … end statement.
   a. MATLAB Program:

```
n = 3
if n>4
    j=10;
else
    j=12;
end
j
```

   b. Output:

```
n =

     3


j =

    12
```

c) if … elseif … else … end statement.

a. MATLAB Program:

```
x = 10;
minVal = 2;
maxVal = 6;

if (x >= minVal) && (x <= maxVal)
    disp('Value within specified range.')
elseif (x > maxVal)
    disp('Value exceeds maximum value.')
else
    disp('Value is below minimum value.')
end
```

b. Output:

```
Value exceeds maximum value.
```

d) The nested if statements.

a. MATLAB Program:

```
x = 10;
y = 5;
if x == 10
    disp('Outer condition true')
    if(y == 5)
        disp('inner condition true')
    end
else
    disp('Outer condition failed')
end
```

b. Output:

```
Outer condition true
inner condition true
```

e) switch statement.

a. MATLAB Program:

```
x = 10;

switch x
    case 1
        disp('One')
    case 5
        disp('Five')
    case 10
        disp('Ten')
    otherwise
        disp('Unknown value')
end
```

b. Output:

```
Ten
```

f) Nested switch statements.

a. MATLAB Program:

```
x = 10;
y = 5;
switch x
    case 10
        disp('This is from outer switch')
        switch y
            case 5
                disp('This is from inner switch')
            otherwise
                disp('Unknown y value')
        end

    otherwise
        disp('Unknown x value')
end
```

b. Output:

```
This is from outer switch
This is from inner switch
```