MAT 2002

MATLAB

Final Assessment Test

November 6, 2020

SET C

L29+L30

FALL SEMESTER 2020-21

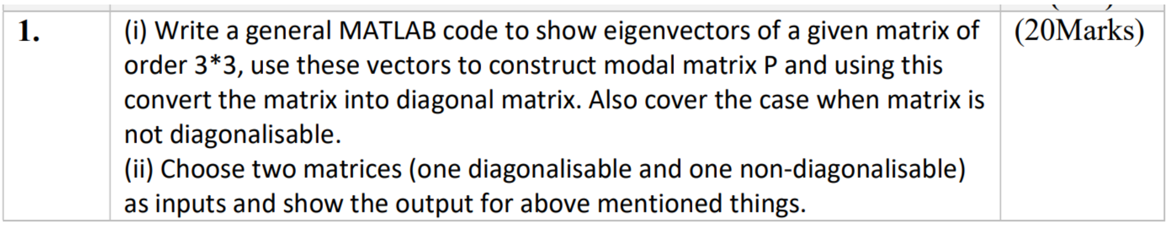
by

SHARADINDU ADHIKARI

19BCE2105

Question 1

Problem:



1.	(i) Write a general MATLAB code to show eigenvectors of a given matrix of order 3*3, use these vectors to construct modal matrix P and using this convert the matrix into diagonal matrix. Also cover the case when matrix is not diagonalisable. (ii) Choose two matrices (one diagonalisable and one non-diagonalisable) as inputs and show the output for above mentioned things.	(20Marks)

Solutions:

Procedure:

- Input the matrix A
- Calculate the eigenvalues and eigenvectors
- Using the matlab command eig(A) calculate the modal matrix P
- If eigenvalues are all different then diagonalizable else if for same eigen-value we don't get different eigen-vectors then not diagonalizable
- Display the diagonal matrix D

Code in MATLAB:

```
clc
clear
close all

% Taking the matrix to diagonalise from user
input_matrix = input("Enter the 3x3 matrix : ");
% Calculate the eigen values of the matrix
eigen_values = eig(input_matrix);
% Displaying the engen values of the matrix
disp("Eigenvalues of the given matrix are: ");
disp(eigen_values);
% Extracting indiviual eigenvalues into a, b, c
a = eigen_values(1); b = eigen_values(2) ; c = eigen_values(3);
% Finding the modal matrix and diagonal matrix
[modal_matrix D] = eig(input_matrix);
% Outputting the eigen vectors
disp("The eigen-vectors of the matrix are : ");
disp(modal_matrix);
% Check if matrix is diagonalisable or not
if(isequal(modal_matrix(:, 1), modal_matrix(:, 2)) &&
isequal(modal_matrix(:, 2), modal_matrix(:, 3)) &&
isequal(modal_matrix(:, 3), modal_matrix(:, 1))) % If diagonalisable
case
    disp("The given matrix is not diagonalisable.");
else % If not diagonalisable case
    PI = inv(modal_matrix); % Calculating the inverse matrix of P
    disp("The diaognal matrix is : ");
    Da = PI * input_matrix * modal_matrix;
    disp(Da);
end
```

NON DIAGNOLISABLE:

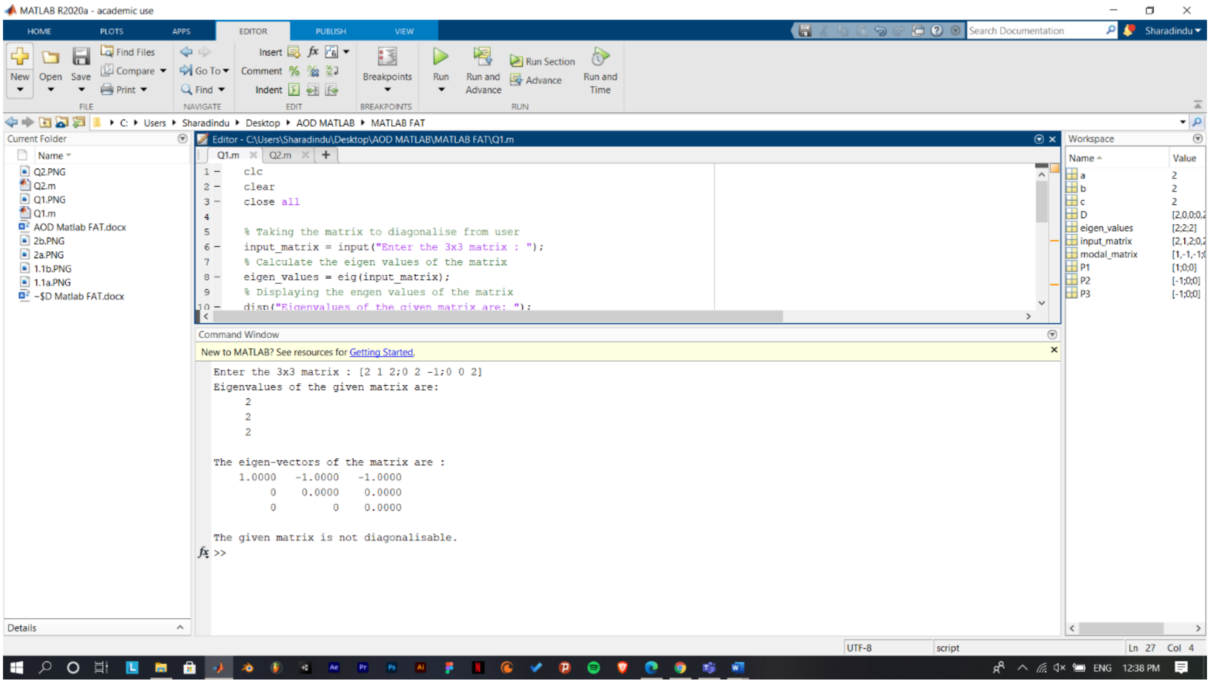
Input in Command Window:

```
Enter the 3x3 matrix : [2 1 2;0 2 -1;0 0 2]
```

Output in Command Window:

```
Eigenvalues of the given matrix are:
     2
     2
     2

The eigen-vectors of the matrix are :
    1.0000   -1.0000   -1.0000
         0    0.0000    0.0000
         0         0    0.0000

The given matrix is not diagonalisable.
```

Screenshot:



```
clc
clear
close all

% Taking the matrix to diagonalise from user
input_matrix = input("Enter the 3x3 matrix : ");
% Calculate the eigen values of the matrix
eigen_values = eig(input_matrix);
% Displaying the engen values of the matrix
disp("Eigenvalues of the given matrix are: ");
```

```
Enter the 3x3 matrix : [2 1 2;0 2 -1;0 0 2]
Eigenvalues of the given matrix are:
     2
     2
     2

The eigen-vectors of the matrix are :
    1.0000   -1.0000   -1.0000
         0    0.0000    0.0000
         0         0    0.0000

The given matrix is not diagonalisable.
>>
```

DIAGNOLISABLE:

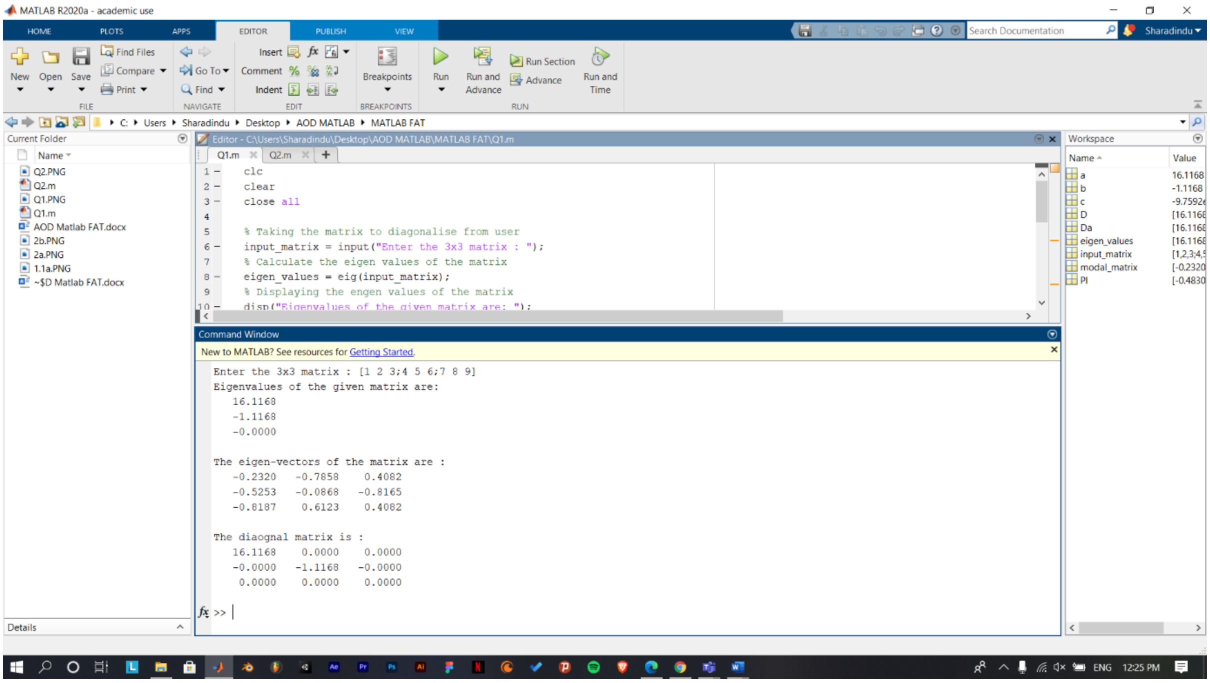
Input in Command Window:

```
Enter the 3x3 matrix : [1 2 3;4 5 6;7 8 9]
```

Output in Command Window:

```
Eigenvalues of the given matrix are:
   16.1168
   -1.1168
   -0.0000

The eigen-vectors of the matrix are :
   -0.2320   -0.7858    0.4082
   -0.5253   -0.0868   -0.8165
   -0.8187    0.6123    0.4082

The diaognal matrix is :
   16.1168    0.0000    0.0000
   -0.0000   -1.1168   -0.0000
    0.0000    0.0000    0.0000
```

Screenshot:

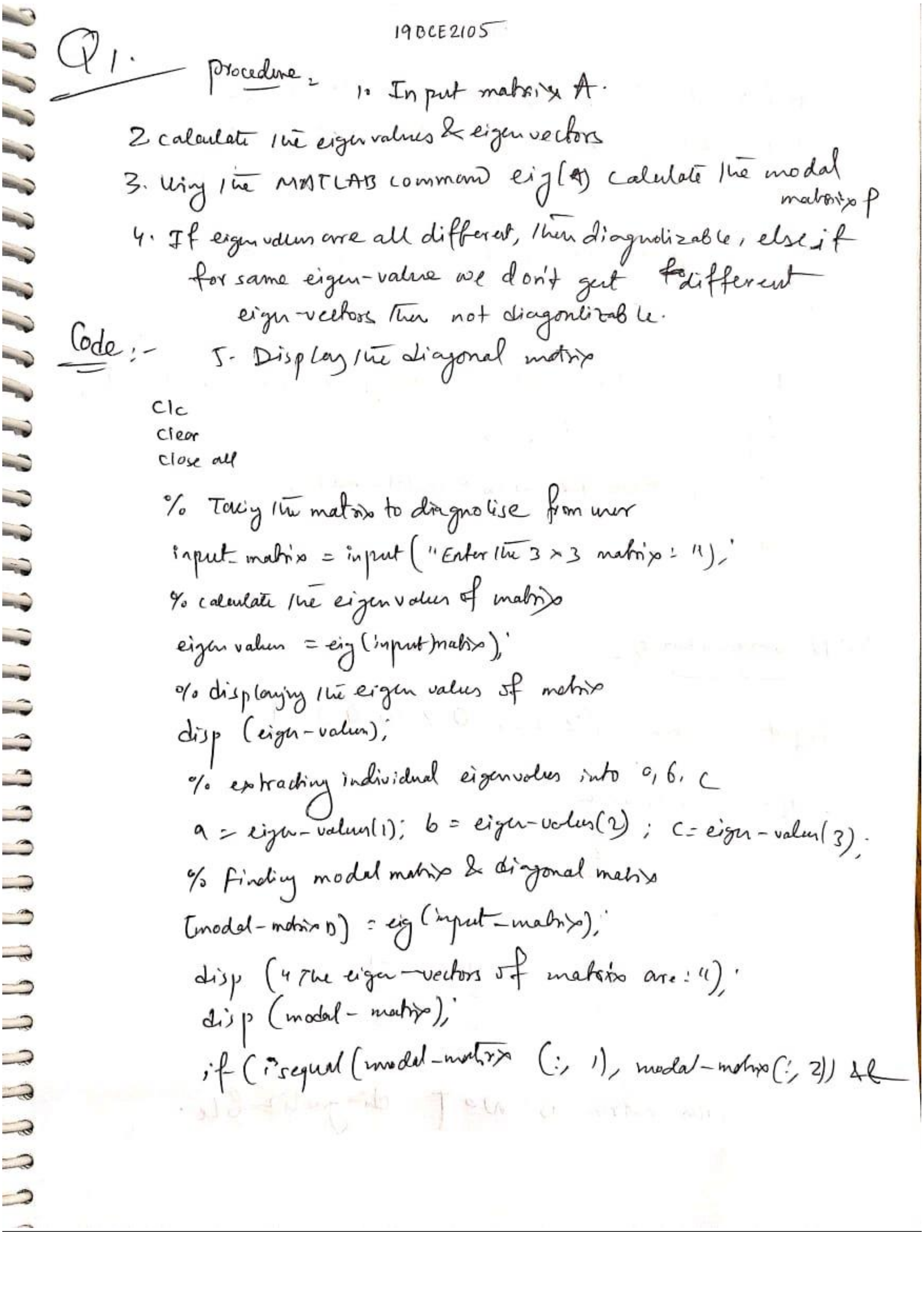


```
clc
clear
close all

% Taking the matrix to diagonalise from user
input_matrix = input("Enter the 3x3 matrix : ");
% Calculate the eigen values of the matrix
eigen_values = eig(input_matrix);
% Displaying the engen values of the matrix
disp("Eigenvalues of the given matrix are: ");
```

```
Enter the 3x3 matrix : [1 2 3;4 5 6;7 8 9]
Eigenvalues of the given matrix are:
   16.1168
   -1.1168
   -0.0000

The eigen-vectors of the matrix are :
   -0.2320   -0.7858    0.4082
   -0.5253   -0.0868   -0.8165
   -0.8187    0.6123    0.4082

The diaognal matrix is :
   16.1168    0.0000    0.0000
   -0.0000   -1.1168   -0.0000
    0.0000    0.0000    0.0000

fx >>
```

Name	Value
a	16.1168
b	-1.1168
c	-9.7592e
D	[16.1168
Da	[16.1168
eigen_values	[16.1168
input_matrix	[1,2,3;4,5
modal_matrix	[-0.2320
PI	[-0.4830

HANDWRITTEN:



19BCE2105

Q1. procedure: 1. Input matrix A.

2 calculate the eigen values & eigen vectors

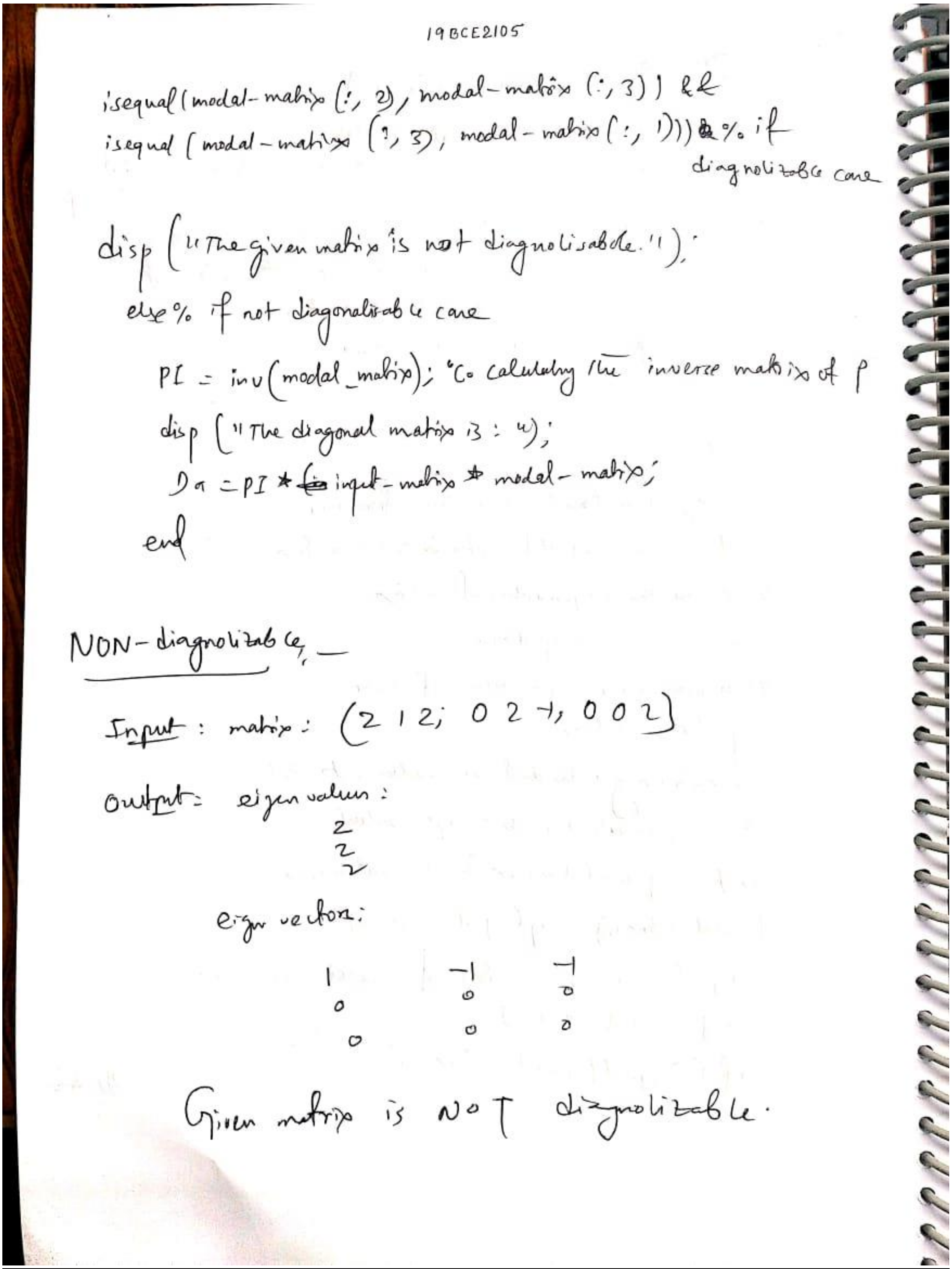
3. Using the MATLAB command eig(A) calculate the modal matrix P

4. If eigen values are all different, then diagonalizable, else if for same eigen-value we don't get different eigen-vectors then not diagonlizable.

Code :-

5. Display the diagonal matrix

```
clc
clear
close all

% Taking the matrix to diagnolise from user
input_matrix = input("Enter the 3 x 3 matrix : ");
% calculate the eigen values of matrix
eigen_values = eig(input_matrix);
% displaying the eigen values of matrix
disp (eigen_values);
 % extracting individual eigenvalues into a, b, c
  a = eigen_value(1); b = eigen_values(2) ; c = eigen_value(3);
 % Finding modal matrix & diagonal matrix
[modal_matrix D] = eig(input_matrix);
 disp ("The eigen-vectors of matrix are : ");
 disp (modal_matrix);
 if (isequal(modal_matrix (:, 1), modal_matrix(:, 2)) &&
```



```
isequal(modal-matrix (:, 2), modal-matrix (:, 3)) &&
isequal (modal-matrix (:, 3), modal-matrix (:, 1))) ~~&~~ % if
                                              diagnolizable case

disp ("The given matrix is not diagnolisable.");
  else % if not diagonalisable case
    PI = inv(modal_matrix); % calculating the inverse matrix of P
    disp ("The diagonal matrix is : ");
    Da = PI * ~~(~~ input-matrix * model-matrix;
  end
```

NON-diagnolizable, —

Input : matrix : (2 1 2; 0 2 -1; 0 0 2)

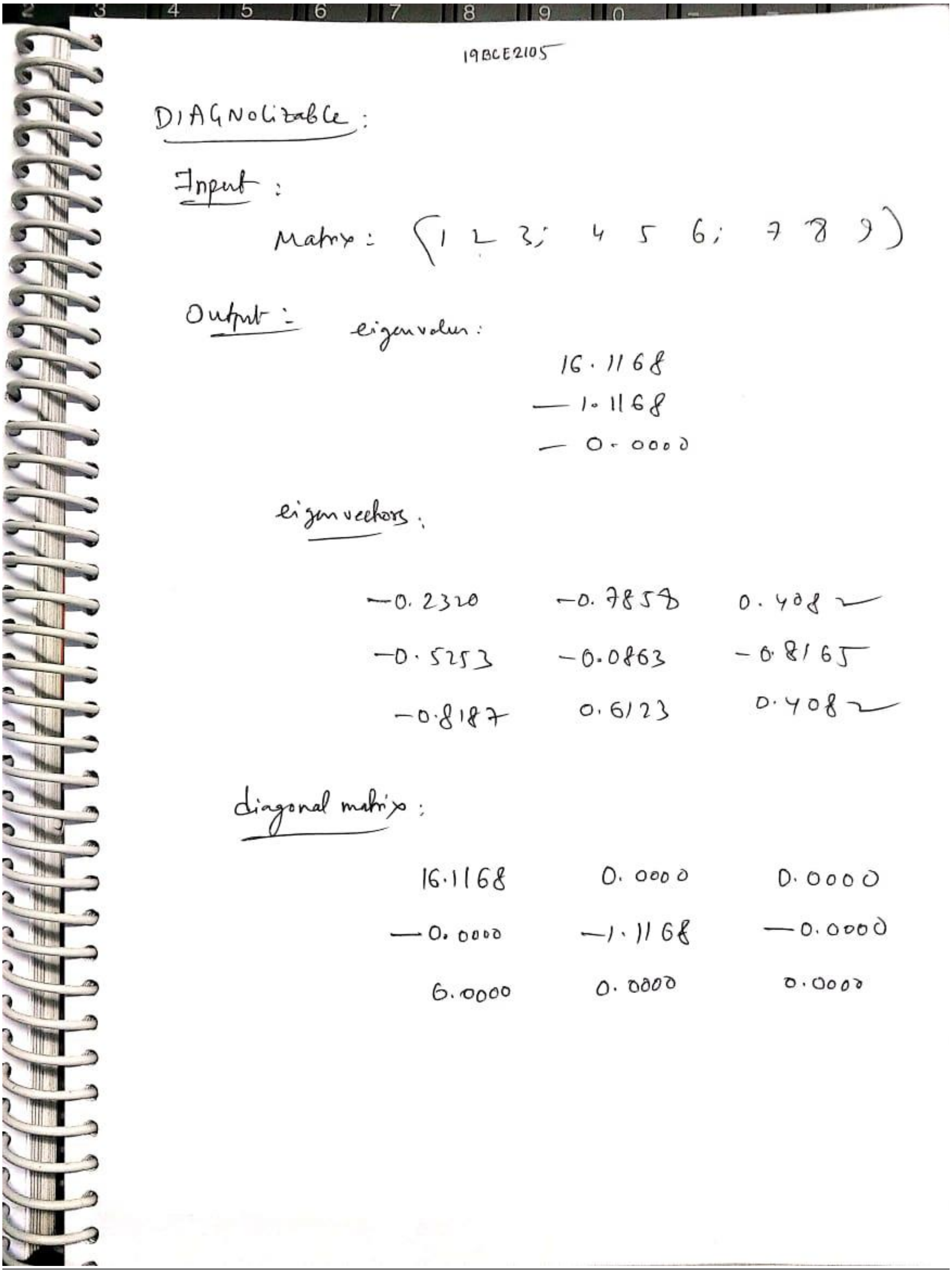
Output: eigen values:

```
        2
        2
        2
```

eigen vectors:

```
   1     -1     -1
   0      0      0
   0      0      0
```

Given matrix is NOT diagnolizable.



19BCE2105

DIAGNOlizable :

Input :

Matrix : [1 2 3; 4 5 6; 7 8 9]

Output : eigenvalues :

```
16.1168
—1.1168
— 0.0000
```

eigenvectors :

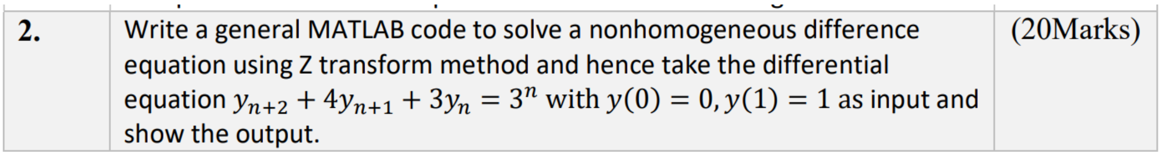
```
—0.2320   —0.7858   0.4082
—0.5253   —0.0863   —0.8165
—0.8187   0.6123    0.4082
```

diagonal matrix :

```
16.1168    0.0000    0.0000
—0.0000   —1.1168   —0.0000
6.0000     0.0000    0.0000
```

Question 2

Problem:



2.	Write a general MATLAB code to solve a nonhomogeneous difference equation using Z transform method and hence take the differential equation $y_{n+2} + 4y_{n+1} + 3y_n = 3^n$ with $y(0) = 0, y(1) = 1$ as input and show the output.	(20Marks)

Solutions:

DIGITAL:

Procedure:

1. Input the difference equation coefficients and the right hand side function of (1).
2. Input the initial conditions (2).
3. Apply Z–Transform and find $Y(z)$.
4. Apply inverse Z – Transform and find y_n .

Code in MATLAB Editor:

```
clc
syms n z y(n) Y
yn=y(n);
yn1=y(n+1);
yn2=y(n+2);
F = input('Input the coefficients [a,b,c]: ');
a=F(1);b=F(2);c=F(3);
nh = input('Enter the non-homogenous part f(n): ');
eqn=a*yn2+b*yn1+c*yn-nh;
ZTY=ztrans(eqn);
IC=input('Enter the initial conditions in the form [y0,y1]:');
y0=IC(1);y1=IC(2);
ZTY=subs(ZTY,{ztrans(y(n),n,z),y(0),y(1)},{Y,y0,y1});
eq=collect(ZTY,Y);
Y=simplify(solve(eq,Y));
yn=simplify(iztrans(Y));
disp('The solution of the difference equation yn=')
disp(yn);
m=0:20;
y=subs(yn,n,m);
stem(y)
title('Difference equation');
xlabel('n'); ylabel('y(n)');
```

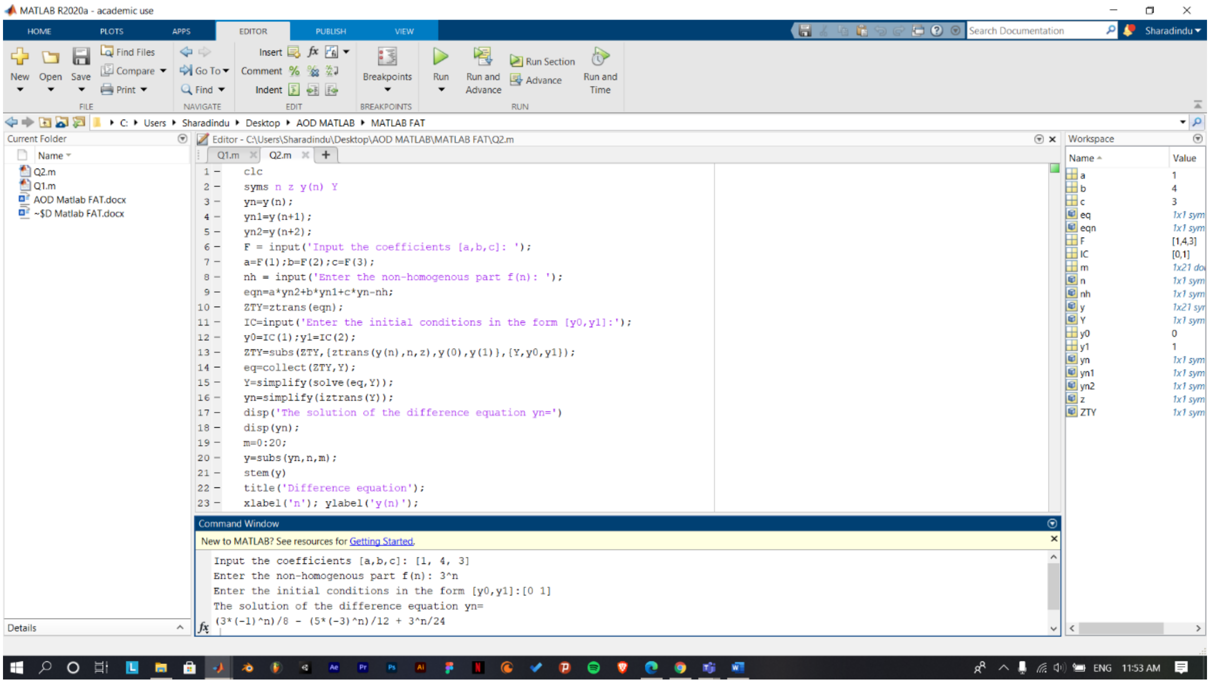
Input in Command Window:

```
Input the coefficients [a,b,c]: [1, 4, 3]
Enter the non-homogenous part f(n): 3^n
Enter the initial conditions in the form [y0,y1]:[0 1]
```

Output in Command Window:

```
The solution of the difference equation yn=
(3*(-1)^n)/8 - (5*(-3)^n)/12 + 3^n/24
```

Screenshot & Graph:

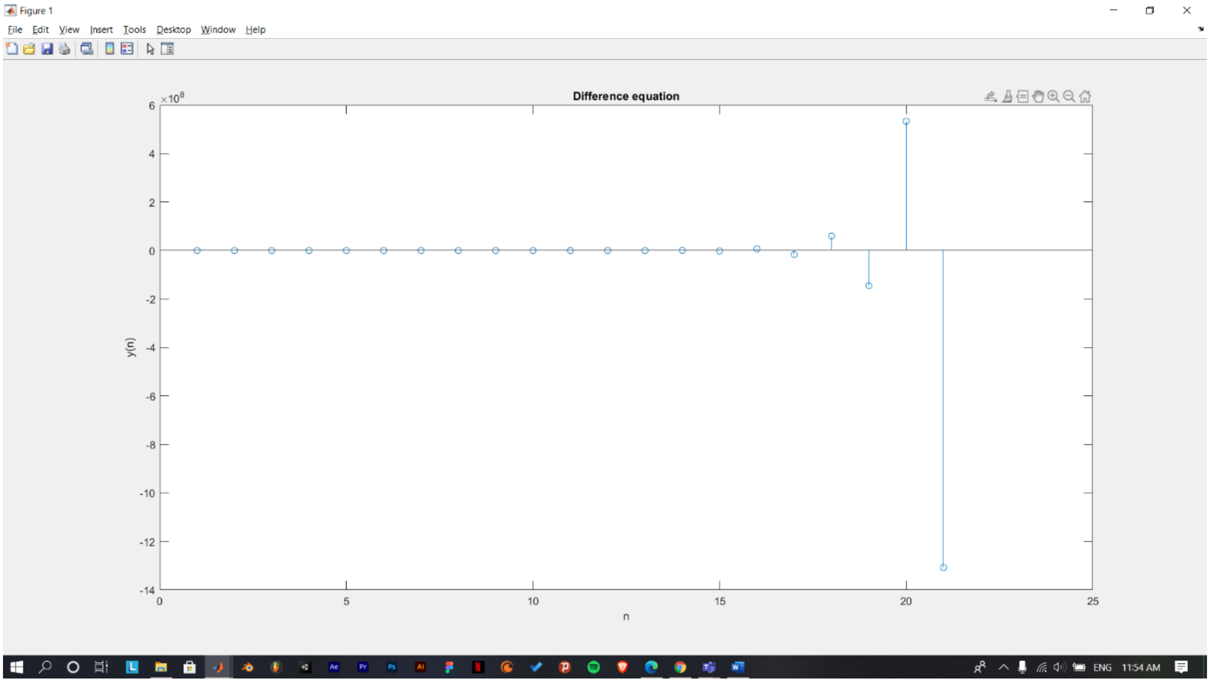


```
clc
syms n z y(n) Y
yn=y(n);
yn1=y(n+1);
yn2=y(n+2);
F = input('Input the coefficients [a,b,c]: ');
a=F(1);b=F(2);c=F(3);
nh = input('Enter the non-homogenous part f(n): ');
eqn=a*yn2+b*yn1+c*yn-nh;
ZTY=ztrans(eqn);
IC=input('Enter the initial conditions in the form [y0,y1]:');
y0=IC(1);y1=IC(2);
ZTY=subs(ZTY,{ztrans(y(n),n,z),y(0),y(1)},{Y,y0,y1});
eq=collect(ZTY,Y);
Y=simplify(solve(eq,Y));
yn=simplify(iztrans(Y));
disp('The solution of the difference equation yn=')
disp(yn);
m=0:20;
y=subs(yn,n,m);
stem(y)
title('Difference equation');
xlabel('n'); ylabel('y(n)');
```

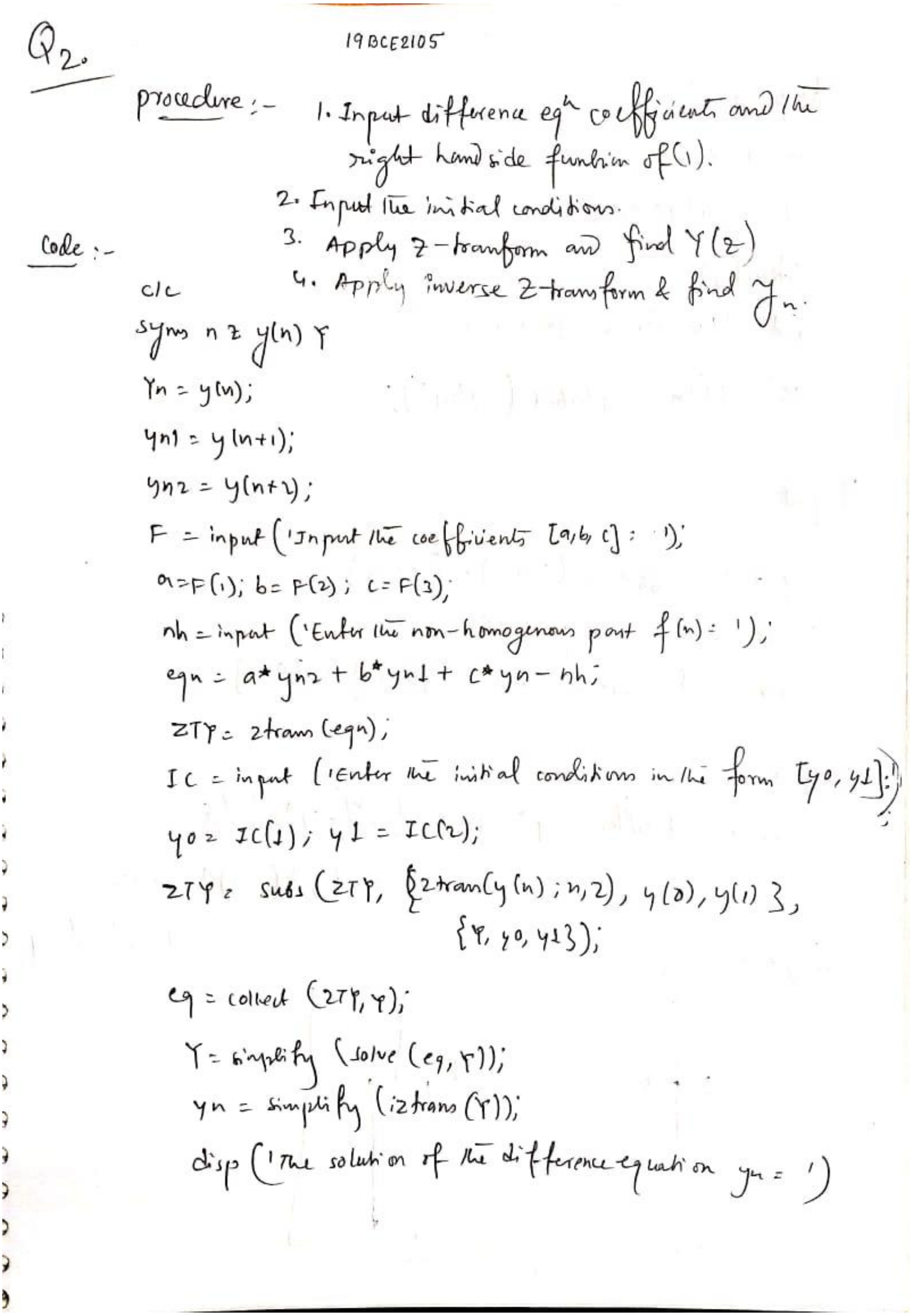
New to MATLAB? See resources for Getting Started.

```
Input the coefficients [a,b,c]: [1, 4, 3]
Enter the non-homogenous part f(n): 3^n
Enter the initial conditions in the form [y0,y1]:[0 1]
The solution of the difference equation yn=
(3*(-1)^n)/8 - (5*(-3)^n)/12 + 3^n/24
```

Name	Value
a	1
b	4
c	3
eq	1x1 sym
eqn	1x1 sym
F	[1,4,3]
IC	[0,1]
m	1x21 do
n	1x1 sym
nh	1x1 sym
y	1x21 sy
Y	1x1 sym
y0	0
y1	1
yn	1x1 sym
yn1	1x1 sym
yn2	1x1 sym
z	1x1 sym
ZTY	1x1 sym



HANDWRITTEN:



Q2.

19BCE2105

procedure :- 1. Input difference eq^n coefficients and the right hand side funtion of(1).

2. Input the initial conditions.

3. Apply Z-tranform and find Y(z)

4. Apply inverse Z-transform & find y_n.

Code :-

```
clc
syms n z y(n) Y
Yn = y(n);
yn1 = y(n+1);
yn2 = y(n+2);
F = input('Input the coefficients [a,b,c] : ');
a=F(1); b= F(2); c= F(3);
nh = input ('Enter the non-homogenous part f(n): ');
eqn = a*yn2 + b*yn1 + c*yn - nh;
ZTY = ztrans (eqn);
IC = input ('Enter the initial conditions in the form [y0, y1]:');
y0 = IC(1); y1 = IC(2);
ZTY = subs (ZTY, {ztran(y(n) ; n,z), y(0), y(1)},
                    {Y, y0, y1});
eq = collect (ZTY, Y);
Y = simplify (solve (eq, Y));
yn = simplify (iztrans (Y));
disp ('The solution of the difference equation yn = ')
disp (yn);
 m=0:20;
 y = subs (yn,n,m);
 stem (y)
 title ('Difference equation');
 xlabel ('n'); ylabel ('y(n)');
```

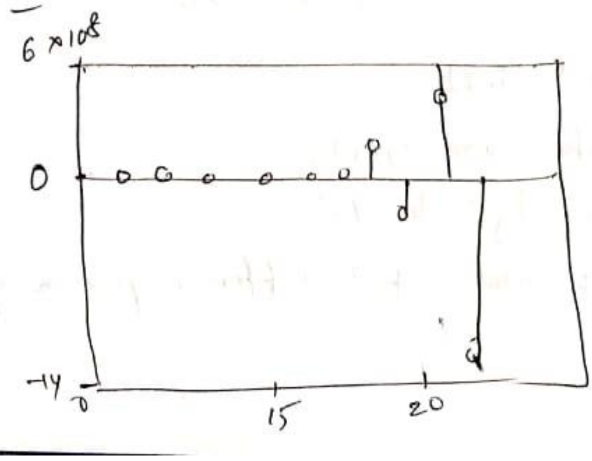
Input :-

```
Input the coefficients [a,b,c] : [1,4,3]
Enter non-homogenous part f(n) : 3^n
Enter initial condition in form [y0,y1] : [0 1]
```

Output:

```
solution of difference eqn  yn =  (3*(-1)^n)/8
                                  - (5*(-3)^n)/12
                                              + 3^n/24
```

Graph :-



6 ×10^8
0
-14
0
15
20