

MATLAB ASSIGNMENT

LINEAR ALGEBRA

UE20MA251

NON IN-BUILT FUNCTION PROGRAMS:

1) GAUSS ELIMINATION METHOD:

$$I) X+2y-z=3, 2x+y-2z=3, -3x+y+z=-6$$

$$C = [1 \ 2 \ -1; \ 2 \ 1 \ -2; \ -3 \ 1 \ 1]$$

$$b = [3 \ 3 \ -6]'$$

$$A = [C \ b];$$

$$n = \text{size}(A, 1);$$

$$x = \text{zeros}(n, 1);$$

$$\text{for } i=1:n-1$$

$$\text{for } j=i+1:n$$

$$m = A(j, i) / A(i, i)$$

$$A(j, :) = A(j, :) - m * A(i, :)$$

end

end

$$x(n) = A(n, n+1) / A(n, n)$$

$$\text{for } i=n-1:-1:1$$

$$\text{summ} = 0$$

$$\text{for } j=i+1:n$$

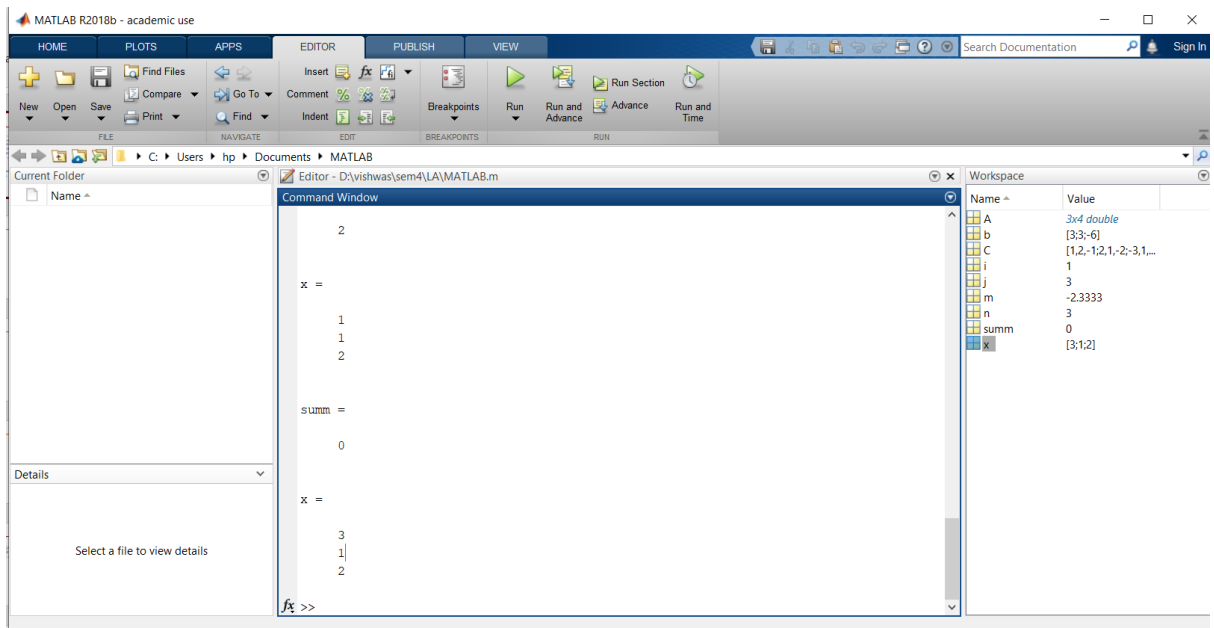
$$\text{summ} = \text{summ} + A(i, j) * x(j, :)$$

$$x(i, :) = (A(i, n+1) - \text{summ}) / A(i, i)$$

end

end

Output:



ii) $x+y+z=11, 2x-6y-1z=0, -3x+4y+2z=-0$

```
C = [1 1 1; 2 -6 -1; 3 4 2]
```

```
b = [11 0 0]'
```

```
A = [C b];
```

```
n = size(A,1);
```

```
x = zeros(n,1);
```

```
for i=1:n-1
```

```
for j=i+1:n
```

```
m = A(j,i)/A(i,i)
```

```
A(j,:) = A(j,:) - m*A(i,:)
```

```
end
```

```
end
```

```
x(n) = A(n,n+1)/A(n,n)
```

```
for i=n-1:-1:1
```

```
summ = 0
```

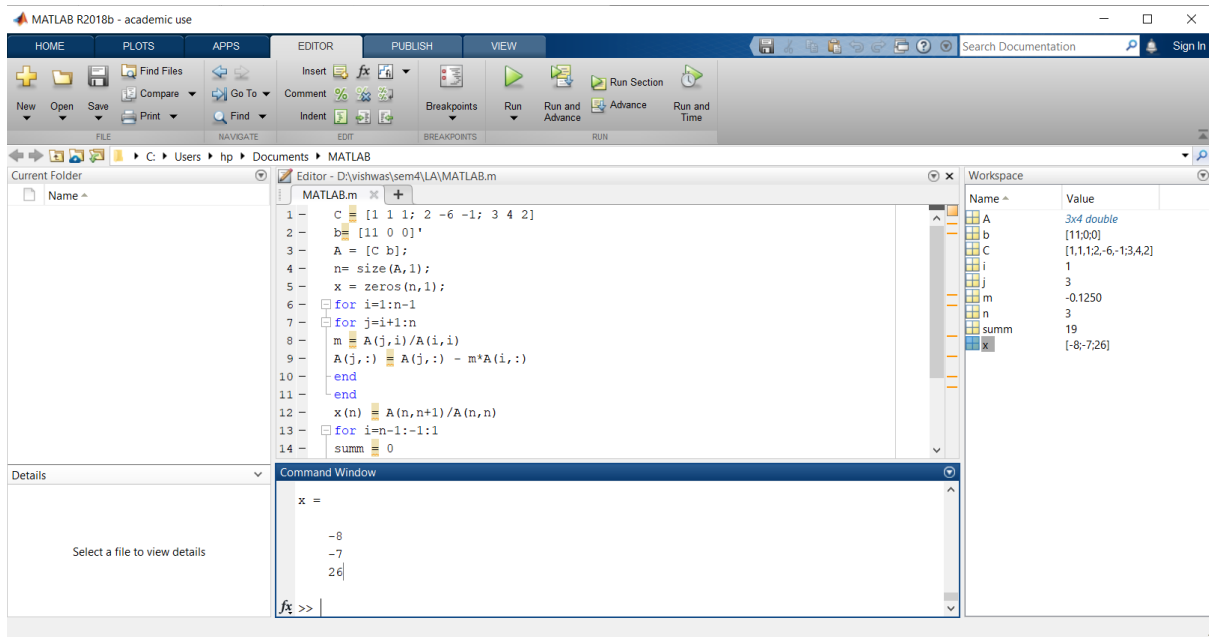
```
for j=i+1:n
```

```

summ = summ + A(i,j)*x(j,:)
x(i,:) = (A(i,n+1) - summ)/A(i,i)
end
end

```

Output :



iii) $2x+y-z=0$, $2x+5y+7z=52$, $x+y+z=-9$

```

C = [2 1 -1; 2 5 7; 1 1 1]
b = [0 52 9]'
A = [C b];
n = size(A,1);
x = zeros(n,1);
for i=1:n-1
    for j=i+1:n
        m = A(j,i)/A(i,i)
        A(j,:) = A(j,:) - m*A(i,:)
    end
end
x(n) = A(n,n+1)/A(n,n)
for i=n-1:-1:1
    summ = 0

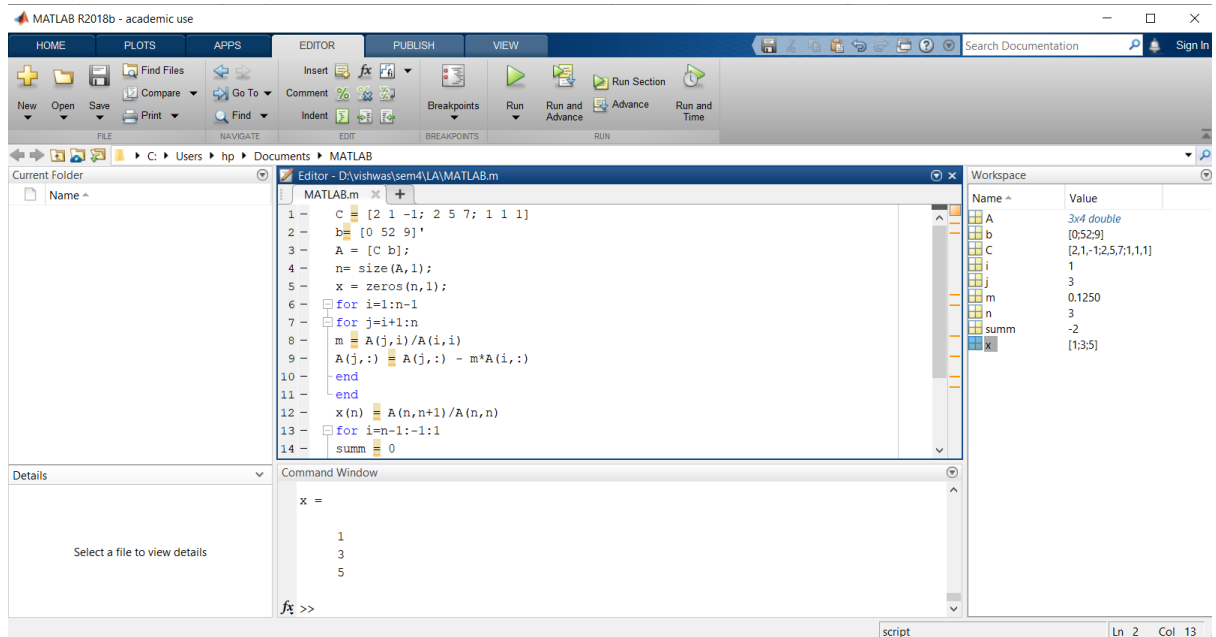
```

```

for j=i+1:n
summ = summ + A(i,j)*x(j,:)
x(i,:) = (A(i,n+1) - summ)/A(i,i)
end
end

```

Output:



2) Gauss - Jordan Method To find Inverse:

1) Find by Gauss Jordan Method $A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}$

```
A = [1, 1, 1; 4, 3, -1; 3, 5, 3];
```

```
n = length(A(1,:));
```

```
Aug = [A, eye(n,n)]
```

```
for j=1:n-1
```

```
for i=j+1:n
```

```
Aug(i,j:2*n) = Aug(i,j:2*n) -
```

```
Aug(i,j)/Aug(j,j)*Aug(j,j:2*n)
```

```
end
```

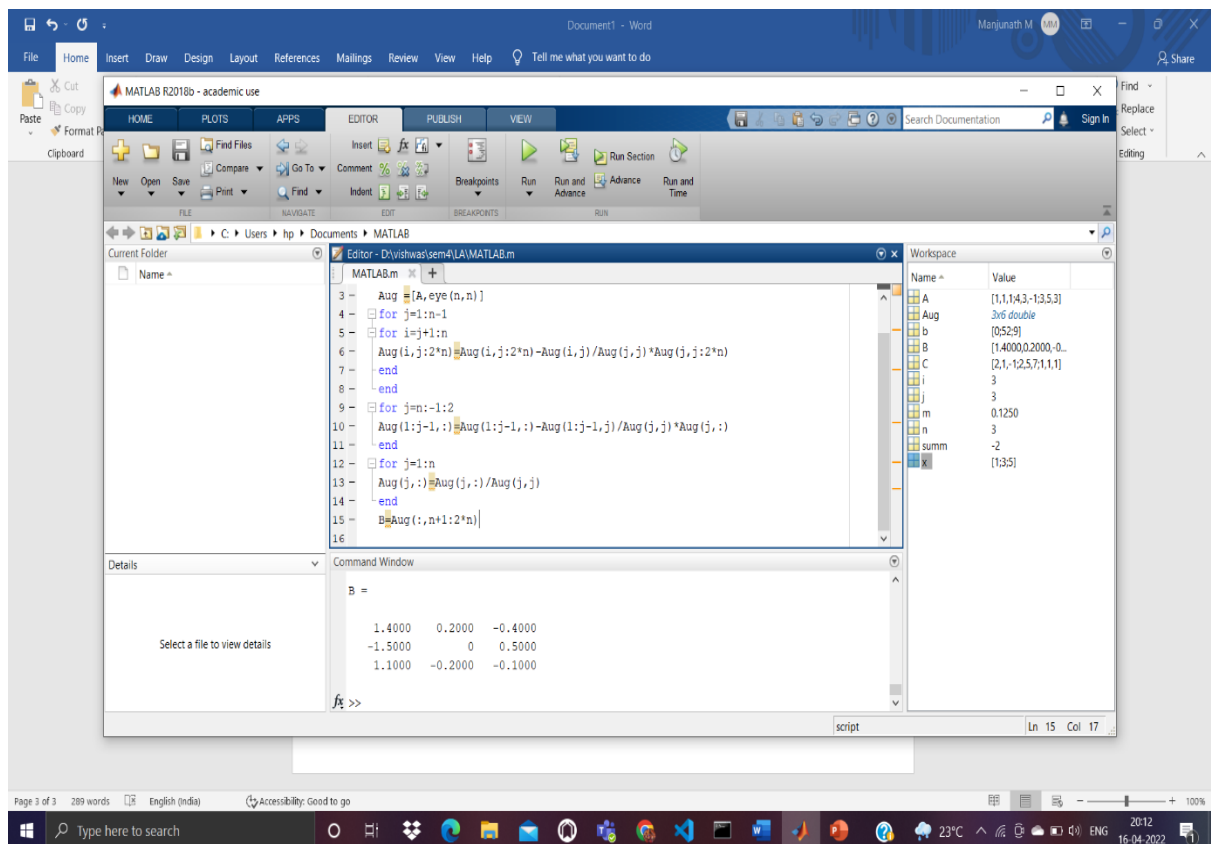
```
end
```

```
for j=n:-1:2
```

```

Aug(1:j1,:) = Aug(1:j1,:) * Aug(1:j1,j) / Aug(j,j)
end
for j=1:n
Aug(j,:) = Aug(j,:) / Aug(j,j)
end
B=Aug(:,n+1:2*n)
Output:

```



ii) Find by Gauss Jordan Method $A = \begin{bmatrix} 1 & 4 & 1 \\ 1 & 2 & 1 \\ 1 & 8 & 5 \end{bmatrix}$

```

A = [1, 4, 1; 1, 2, 1; 1, 8, 5];
n = length(A(1,:));
Aug = [A, eye(n,n)]
for j=1:n-1
for i=j+1:n

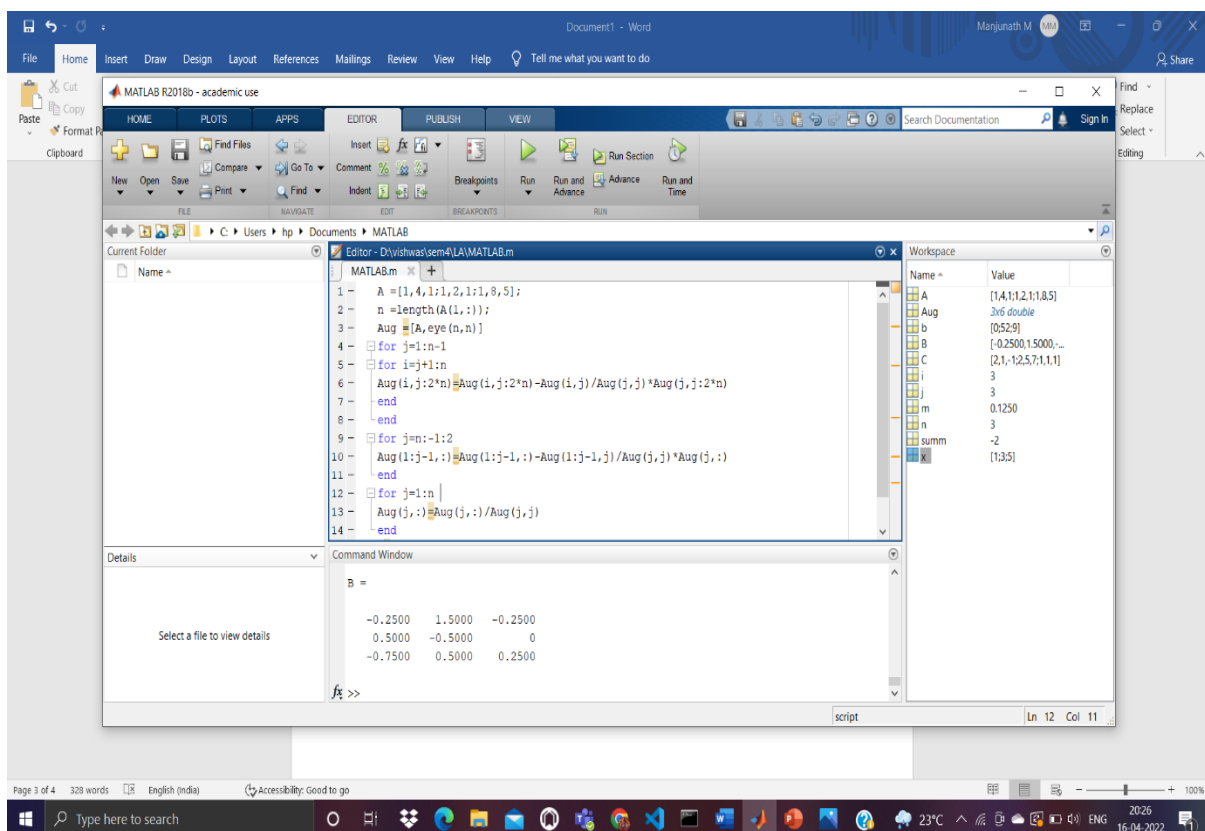
```

```

Aug(i,j:2*n)=Aug(i,j:2*n)-
Aug(i,j)/Aug(j,j)*Aug(j,j:2*n)
end
end
for j=n:-1:2
Aug(1:j1,:)=Aug(1:j1,:)Aug(1:j1,j)/Aug
(j,j)*Aug(j,:)
end
for j=1:n
Aug(j,:)=Aug(j,:)/Aug(j,j)
end
B=Aug(:,n+1:2*n)

```

Output:



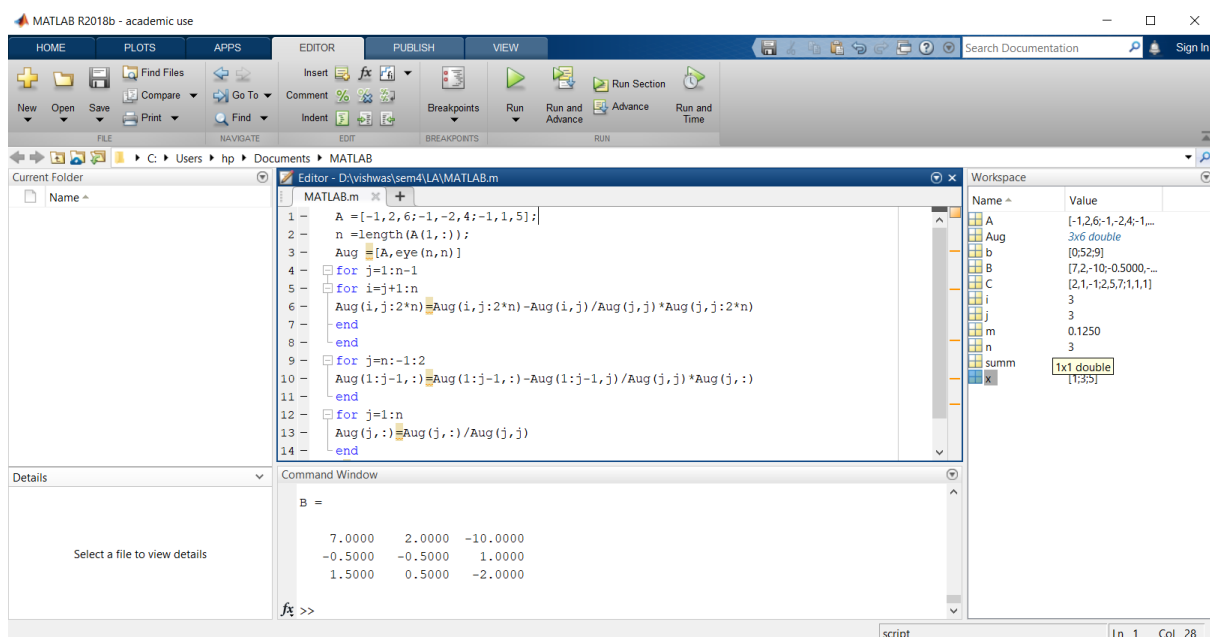
iii) Find by Gauss Jordan Method $A = \begin{bmatrix} -1 & 2 & 6 \\ -1 & -2 & 4 \\ -1 & 1 & 5 \end{bmatrix}$

```

A =[-1, 2, 6;-1, -2, 4;-1, 1, 5];
n =length(A(1,:));
Aug =[A,eye(n,n)]
for j=1:n-1
for i=j+1:n
Aug(i,j:2*n)=Aug(i,j:2*n)-
Aug(i,j)/Aug(j,j)*Aug(j,j:2*n)
end
end
for j=n:-1:2
Aug(1:j-1,:)=Aug(1:j-1,:)-Aug(1:j-1,j)/Aug
(j,j)*Aug(j,:)
end
for j=1:n
Aug(j,:)=Aug(j,:)/Aug(j,j)
end
B=Aug(:,n+1:2*n)

```

Output:



3) LU Decomposition Method:

$$1) A = \begin{bmatrix} 1 & 1 & -1 \\ 3 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
Ab = [1 1 -1;3 5 6;7 8 9];
```

```
n= length(A);
```

```
L = eye(n);
```

```
for i =2:3
```

```
alpha = Ab(i,1)/Ab(1,1);
```

```
L(i,1) = alpha;
```

```
Ab(i,:) = Ab(i,:) - alpha*Ab(1,:);
```

```
end
```

```
i=3;
```

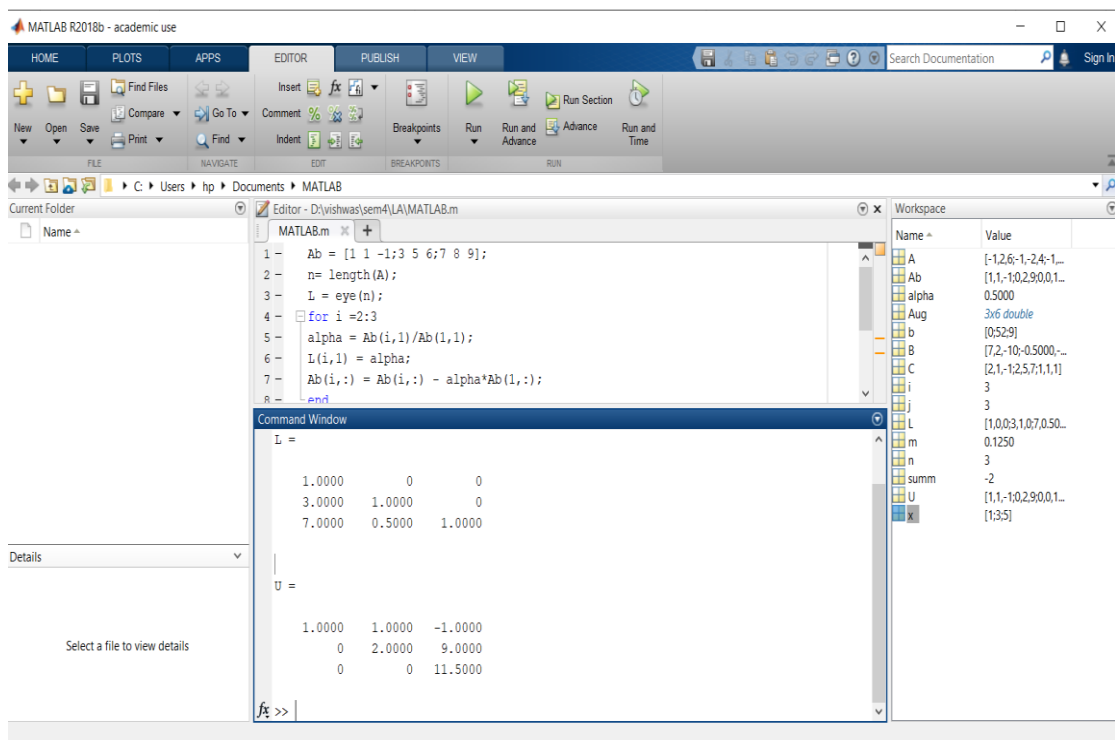
```
alpha = Ab(i,2)/Ab(2,2);
```

```
L(i,2) = alpha
```

```
Ab(i,:) = Ab(i,:) - alpha*Ab(2,:);
```

```
U = Ab(1:n,1:n)
```

Output:



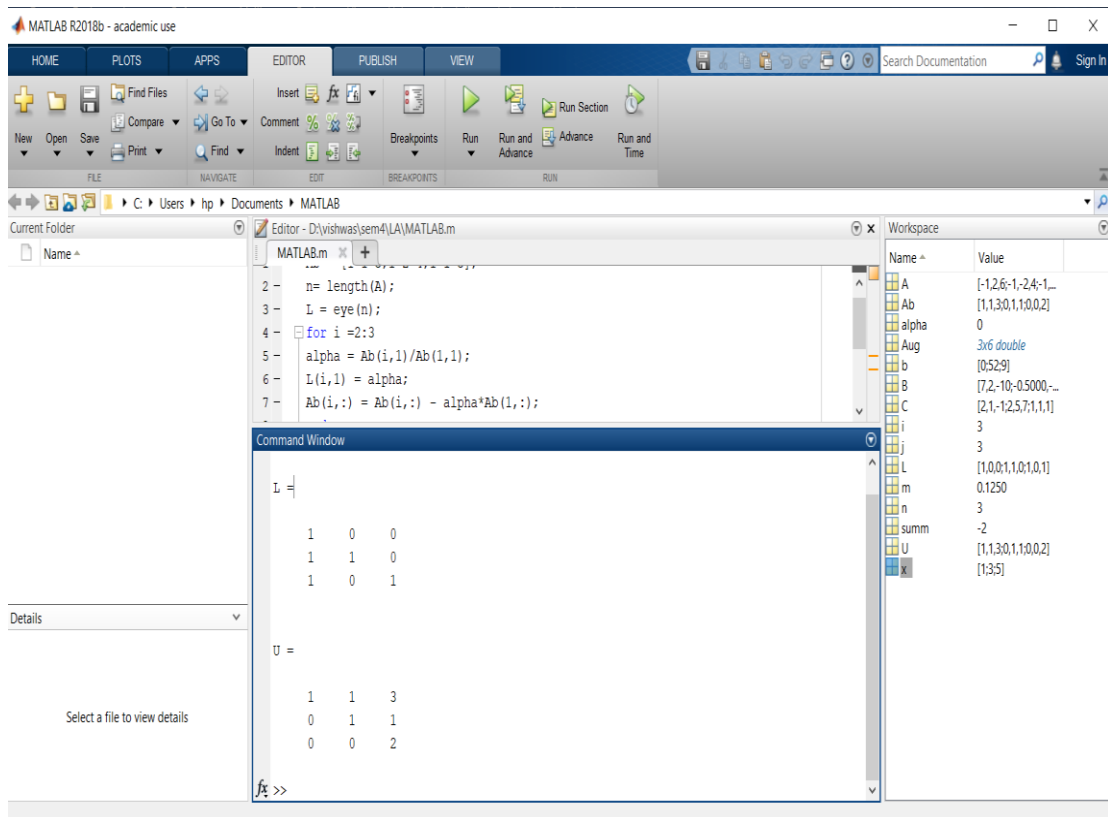
ii) $A = \begin{bmatrix} -1 & 2 & 6 \\ -1 & -2 & 4 \\ -1 & 1 & 5 \end{bmatrix}$

```

Ab = [1 1 3;1 2 4;1 1 5];
n= length(A);
L = eye(n);
for i =2:3
    alpha = Ab(i,1)/Ab(1,1);
    L(i,1) = alpha;
    Ab(i,:) = Ab(i,:) - alpha*Ab(1,:);
end
i=3;
alpha = Ab(i,2)/Ab(2,2);
L(i,2) = alpha
Ab(i,:) = Ab(i,:) - alpha*Ab(2,:);
U = Ab(1:n,1:n)

```

Output:



$$\text{iii) } A = \begin{bmatrix} -1 & 4 & 6 \\ 0 & -2 & 4 \\ 0 & 0 & 5 \end{bmatrix}$$

```

Ab = [-1 4 6;0 -2 4;0 0 5];
n= length(A);
L = eye(n);
for i =2:3
    alpha = Ab(i,1)/Ab(1,1);
    L(i,1) = alpha;
    Ab(i,:) = Ab(i,:) - alpha*Ab(1,:);
end
i=3;
alpha = Ab(i,2)/Ab(2,2);
L(i,2) = alpha
Ab(i,:) = Ab(i,:) - alpha*Ab(2,:);
U = Ab(1:n,1:n)

```

Output:

The screenshot shows the MATLAB R2018b interface. The Editor window displays a script named MATLAB.m with the following code:

```
1 Ab = [-1 4 6; 0 -2 4; 0 0 5];
2 n = length(Ab);
3 L = eye(n);
4 for i = 2:3
5     alpha = Ab(i,1)/Ab(1,1);
6     L(i,1) = alpha;
7     Ab(i,:) = Ab(i,:) - alpha*Ab(1,:);
8 end
9 i=3;
10 alpha = Ab(i,2)/Ab(2,2);
11 L(i,2) = alpha;
12 Ab(i,:) = Ab(i,:) - alpha*Ab(2,:);
13 U = Ab(1:n,1:n)
```

The Command Window shows the output of the script:

```
L =
     1     0     0
     0     1     0
     0     0     1

U =
    -1     4     6
     0    -2     4
     0     0     5
```

The Workspace window shows the following variables:

Name	Value
A	[1, -10.290, 0.6...]
Ab	[-1.460, -2.40, 0.5]
alpha	0
i	3
L	[1.0, 0.0, 1.0, 0.0, 1]
n	3
U	[-1.460, -2.40, 0.5]

4) Grams- Schmidt Orthogonalization process:

- i) Apply the Gram-Schmidt process to the vectors (1,0,1), (1,0,0) and (2,1,0) to produce a set of Orthonormal vectors.

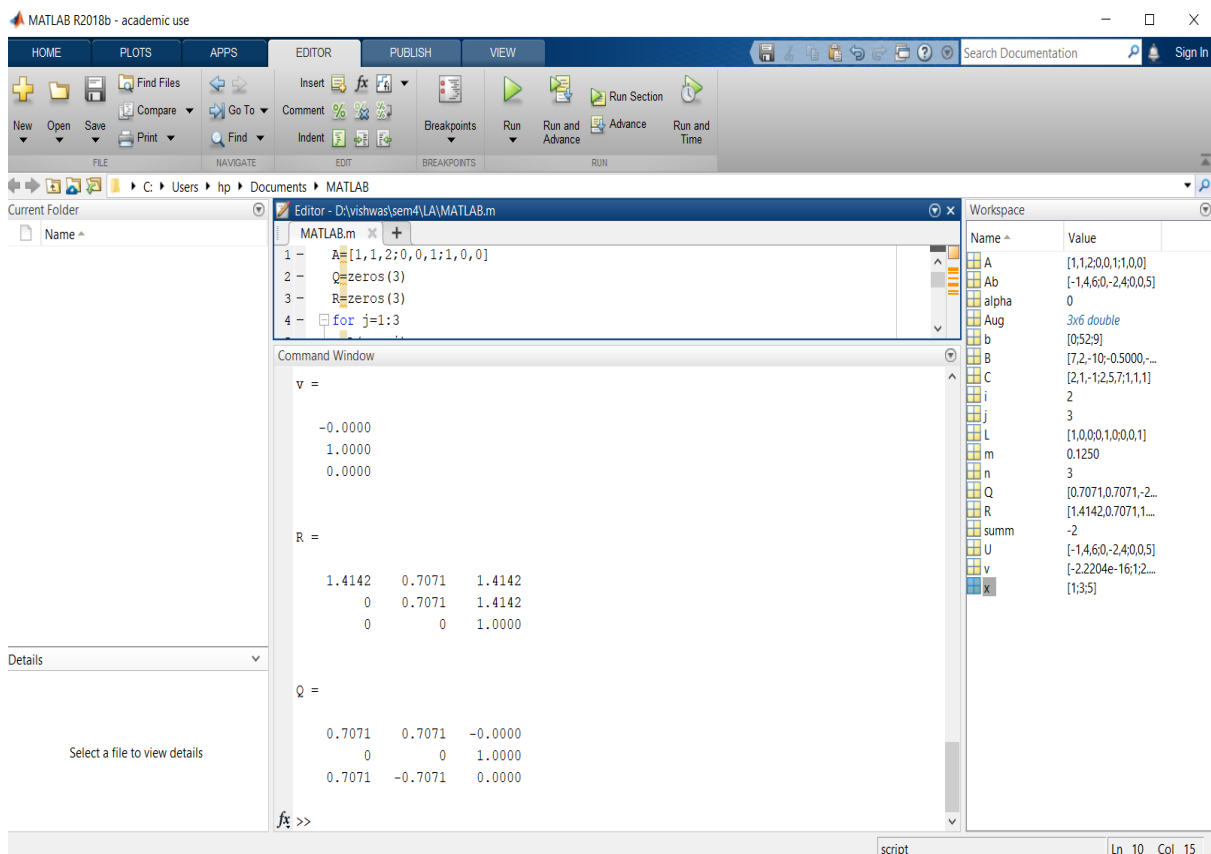
$$A = [1, 1, 2; 0, 0, 1; 1, 0, 0]$$
$$Q = \text{zeros}(3)$$
$$R = \text{zeros}(3)$$

```

for j=1:3
v=A(:,j)
for i=1:j-1
R(i,j)=Q(:,i)'*A(:,j)
v=v-R(i,j)*Q(:,i)
end
R(j,j)=norm(v)
Q(:,j)=v/R(j,j)
end

```

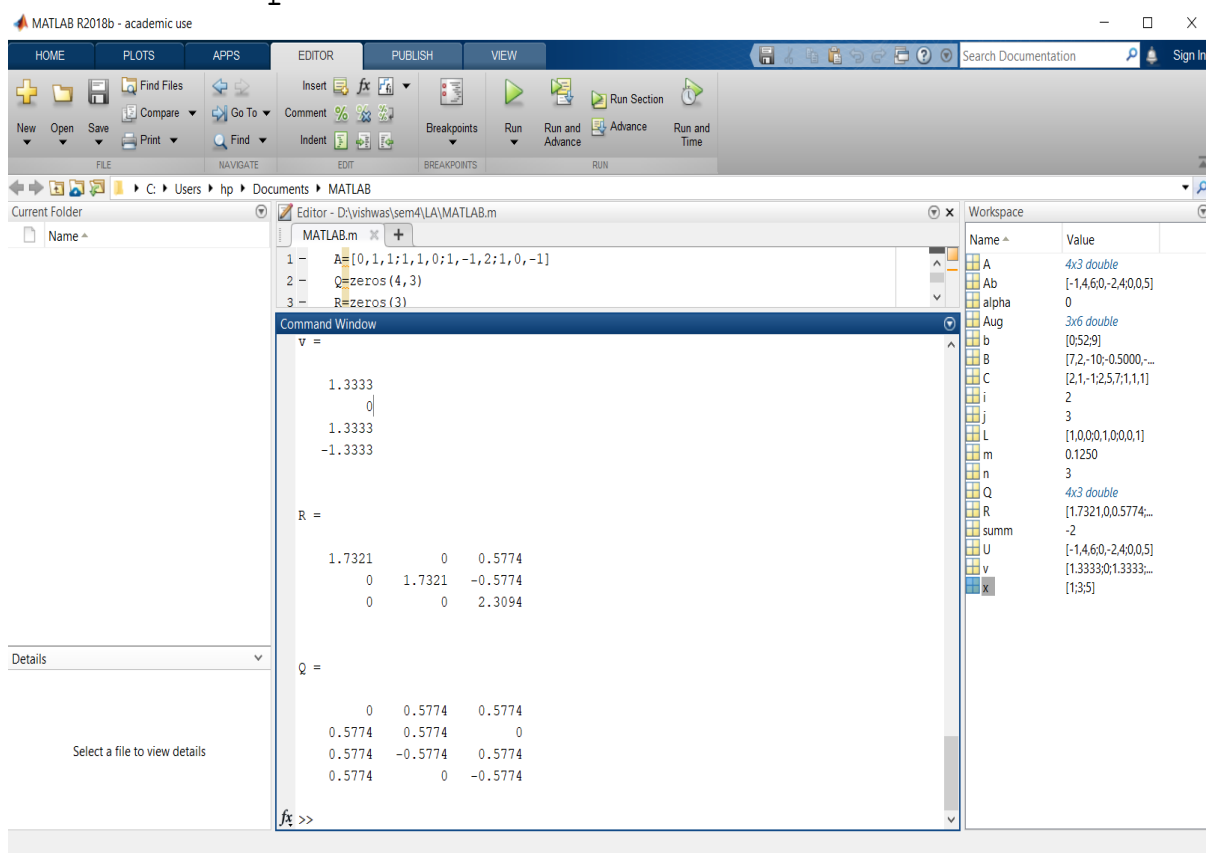
Output:



ii) Apply the Gram-Schmidt process to the vectors $a=(0,1,1,1)$, $b=(1,1,-1,0)$ and $c=(1,0,2,-1)$.

```
A=[0,1,1;1,1,0;1,-1,2;1,0,-1]
Q=zeros(4,3)
R=zeros(3)
for j=1:3
v=A(:,j)
for i=1:j-1
R(i,j)=Q(:,i)'*A(:,j)
v=v-R(i,j)*Q(:,i)
end
R(j,j)=norm(v)
Q(:,j)=v/R(j,j)
end
```

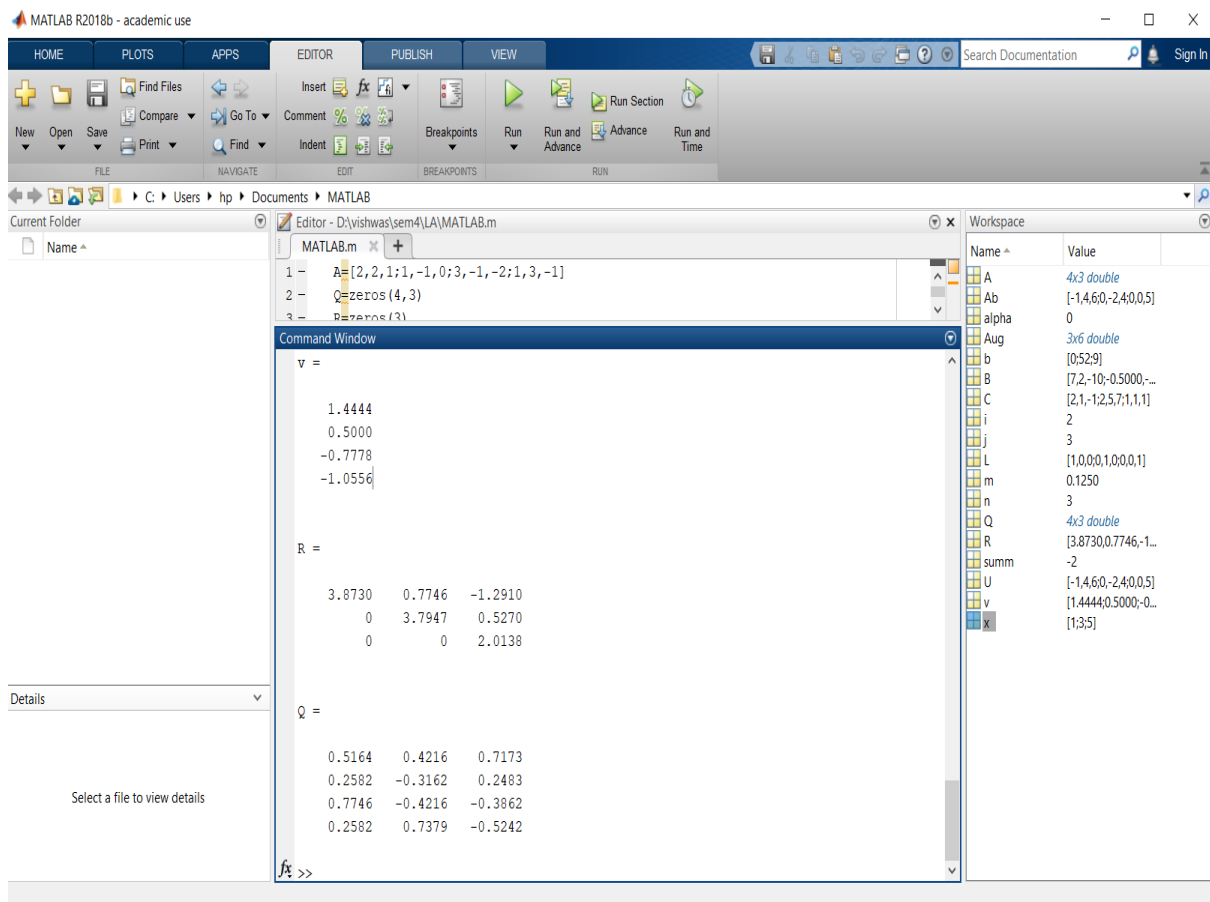
Output:



iii) Apply the Gram-Schmidt process to the vectors $a=(0,1,1,1)$, $b=(1,1,-1,0)$ and $c=(1,0,2,-1)$.

```
A=[2,2,1;1,-1,0;3,-1,-2;1,3,-1]
Q=zeros(4,3)
R=zeros(3)
for j=1:3
    v=A(:,j)
    for i=1:j-1
        R(i,j)=Q(:,i)'*A(:,j)
        v=v-R(i,j)*Q(:,i)
    end
    R(j,j)=norm(v)
    Q(:,j)=v/R(j,j)
end
```

Output:



In-Built Functions:

1) Projection matrices:

i) Find the projection for the matrix $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$

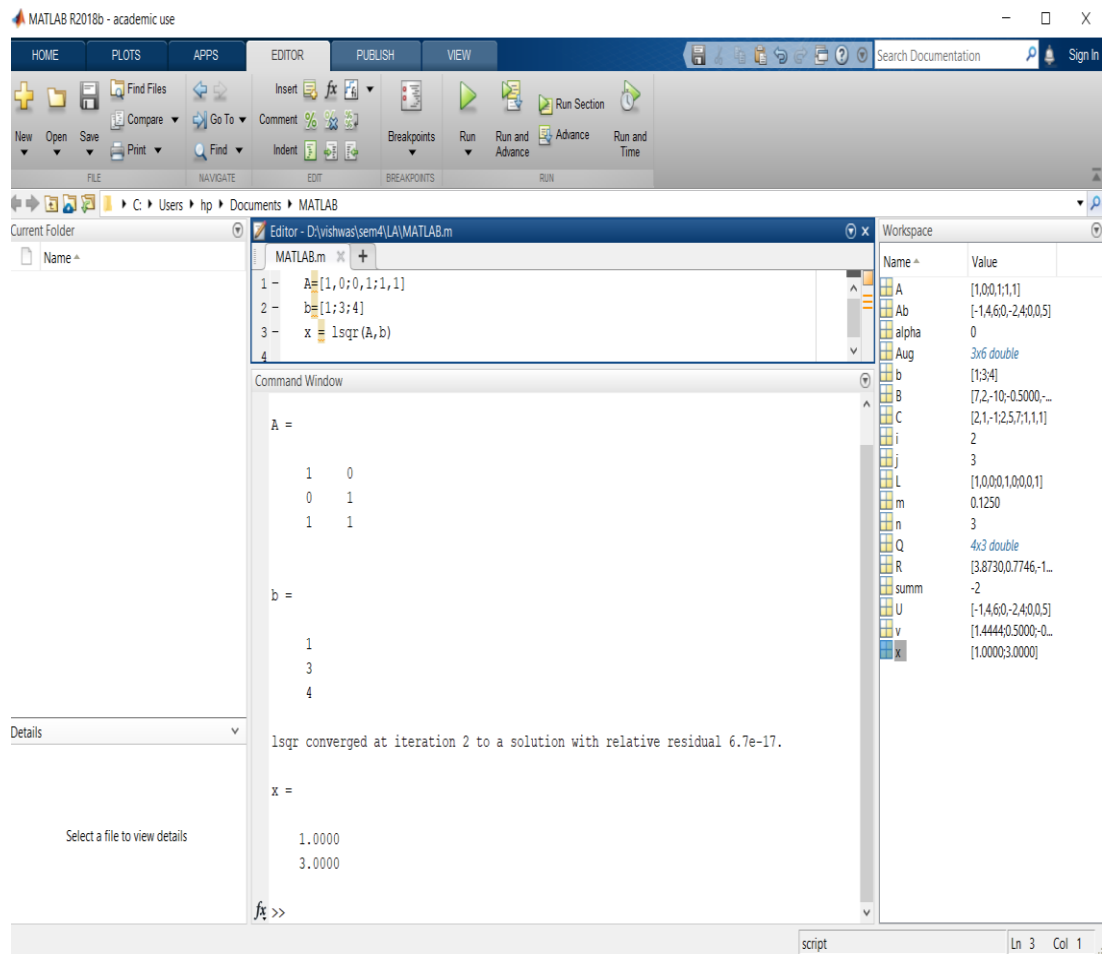
$$u = \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = [1, 0; 0, 1; 1, 1]$$

$$b = [1; 3; 4]$$

$$x = \text{lsqr}(A, b)$$

Output:



ii) Find the projection for the matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

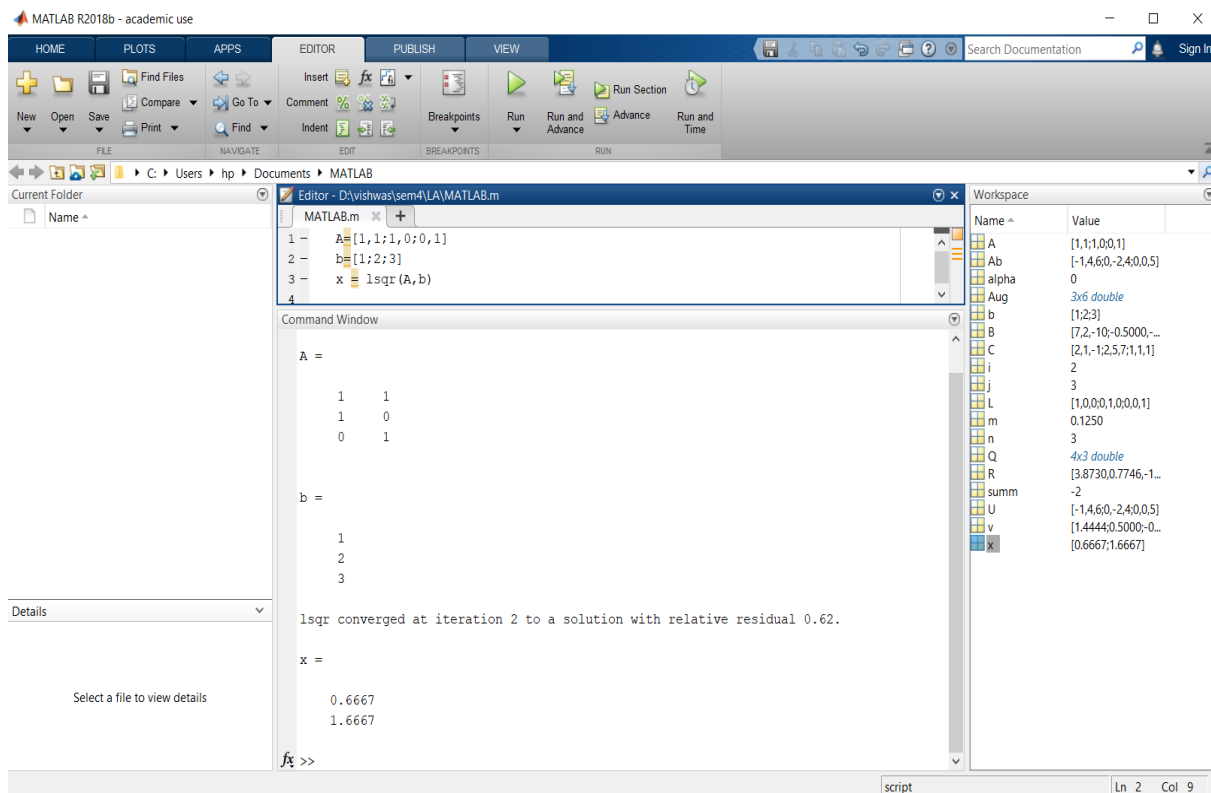
$$u = \begin{bmatrix} u \\ v \end{bmatrix}$$

```

A=[1,1;1,0;0,1]
b=[1;2;3]
x = lsqr(A,b)

```

Output:



iii) Find the projection for the matrix $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$

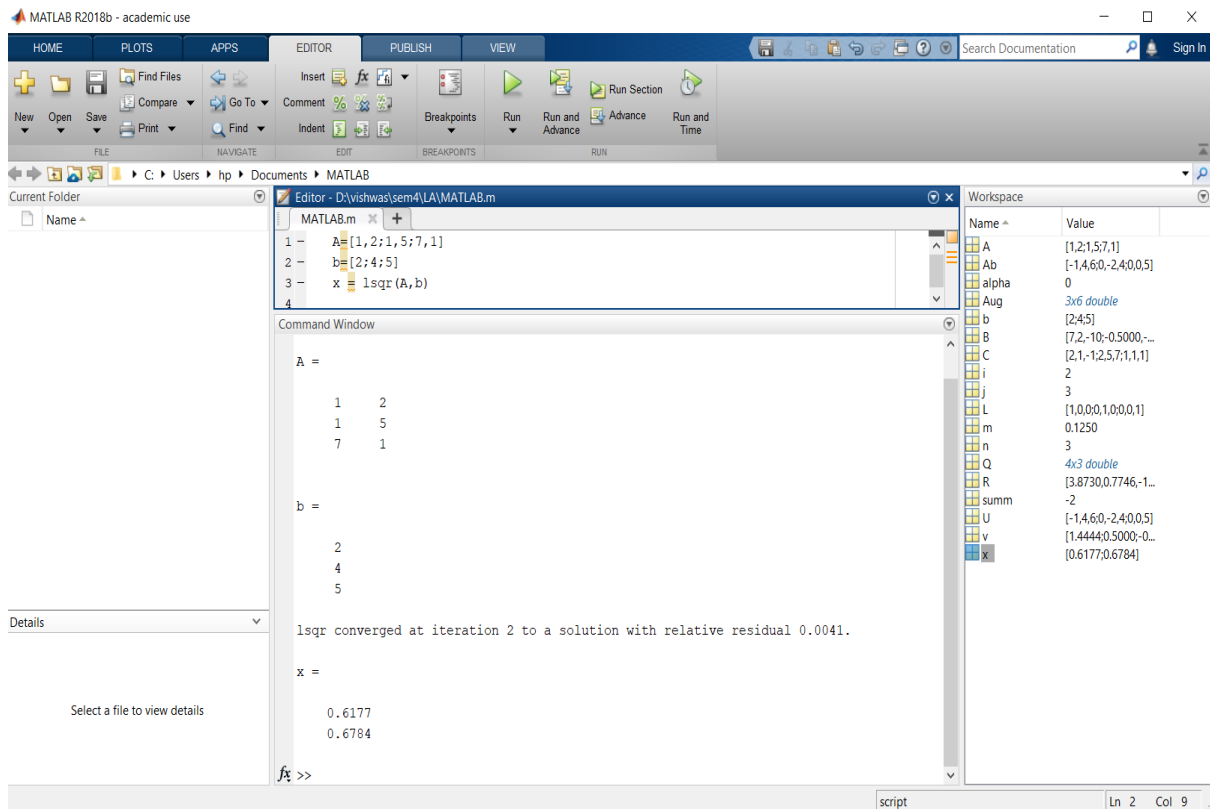
$$u = \begin{bmatrix} u \\ v \end{bmatrix}$$

$$A = [1, 2; 1, 5; 7, 1]$$

$$b = [2; 4; 5]$$

$$x = \text{lsqr}(A, b)$$

Output:



2) least squares:

i) Let $u = \begin{bmatrix} 1 \\ 7 \end{bmatrix}$ onto $v = \begin{bmatrix} -4 \\ 2 \end{bmatrix}$ and find P, the matrix that will project any matrix onto the vector v. Use the result to find $\text{proj}_v u$.

$$u = \begin{bmatrix} 1 \\ 7 \end{bmatrix}$$

$$v = \begin{bmatrix} -4 \\ 2 \end{bmatrix}$$

$$P = (v \cdot \text{transpose}(v)) / (\text{transpose}(v) \cdot v)$$

$$P \cdot u$$

Output:

MATLAB R2018b - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

File Edit Breakpoints Run Run and Advance Run and Time

Current Folder: C:\Users\hp\Documents\MATLAB

Editor: D:\vishwas\sem4\LA\MATLAB.m

```

1 u=[1;7];
2 v=[-4;2];
3 P=(v*transpose(v))/(transpose(v)*v);
4 P*u
5
6

```

Command Window

```

>> MATLAB

P =

    0.8000   -0.4000
   -0.4000    0.2000

ans =

   -2
    1
fx >>

```

Workspace

Name	Value
A	[1,2,1,5;7,1]
Ab	[-1,4,6,0;-2,4,0,0.5]
alpha	0
ans	[-2;1]
Aug	3x6 double
b	[2;4;5]
B	[7,2;-10;-0.5000;...
C	[2,1;-1,2,5,7;1,1,1]
i	2
j	3
L	[1,0,0,0,1,0,0,0,1]
m	0.1250
n	3
P	[0.8000;-0.4000;...
Q	4x3 double
R	[3.8730,0.7746;-1...
summ	-2
u	[1;7]
U	[-1,4,6,0;-2,4,0,0.5]
v	[-4;2]
x	[0.6177;0.6784]

ii) Let $u = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$ onto $v = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$ and find P , the matrix that will project any matrix onto the vector v . Use the result to find $\text{proj}_v u$.

$$u = \begin{bmatrix} 3 \\ 8 \end{bmatrix};$$

$$v = \begin{bmatrix} 4 \\ -2 \end{bmatrix};$$

$$P = (v \cdot \text{transpose}(v)) / (\text{transpose}(v) \cdot v)$$

$$P \cdot u$$

Output:

MATLAB R2018b - academic use

Editor: D:\wshwas\sem4\LA\MATLAB.m

```

1 u=[3;8];
2 v=[4;-2];
3 P=(v*transpose(v))/(transpose(v)*v)
4 P*u
5
6

```

Command Window

```

>> MATLAB

P =

    0.8000    -0.4000
   -0.4000     0.2000

ans =

   -0.8000
    0.4000

```

Workspace

Name	Value
A	[1,2,1,5,7,1]
Ab	[-1,4,6,0,-2,4,0,0,5]
alpha	0
ans	[-0.8000,0.4000]
Aug	3x6 double
b	[2,4,5]
B	[7,2,-10,-0.5000,-...
C	[2,1,-1,2,5,7,1,1,1]
i	2
j	3
L	[1,0,0,1,0,0,0,1]
m	0.1250
n	3
P	[0.8000,-0.4000,-...
Q	4x3 double
R	[3.8730,0.7746,-1...
summ	-2
u	[3;8]
U	[-1,4,6,0,-2,4,0,0,5]
v	[4;-2]
x	[0.6177;0.6784]

script Ln 6 Col 1

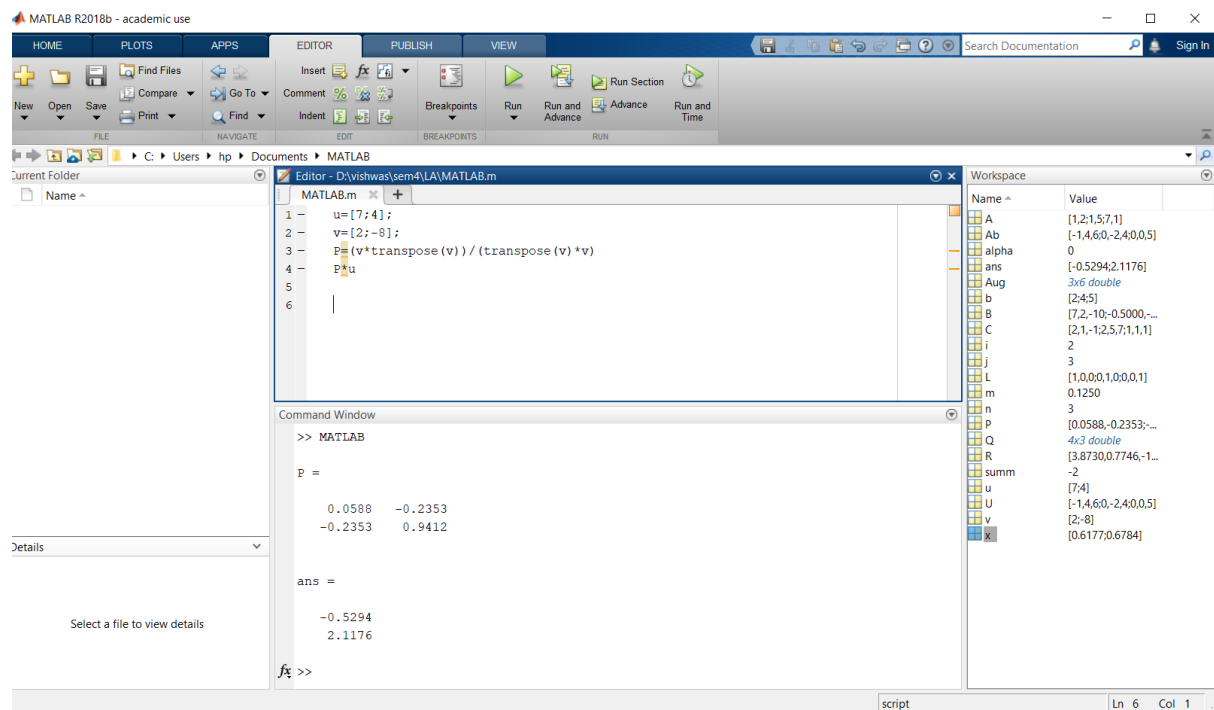
iii) Let $u = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$ onto $v = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$ and find P, the matrix that will project any matrix onto the vector v. Use the result to find proj v u.

```

u=[ 7 ; 4 ] ;
v=[ 2 ; -8 ] ;
P=(v*transpose(v))/(transpose(v)*v)
P*u

```

Output:



3) Eigen values and Eigen vectors:

i) Find the eigenvalues and the corresponding eigenvectors of the matrix $A=[1,1,3;1,5,1;3,1,1]$.

Ans: $A=[1,1,3;1,5,1;3,1,1]$

$e=eig(A)$

$\det(A)$

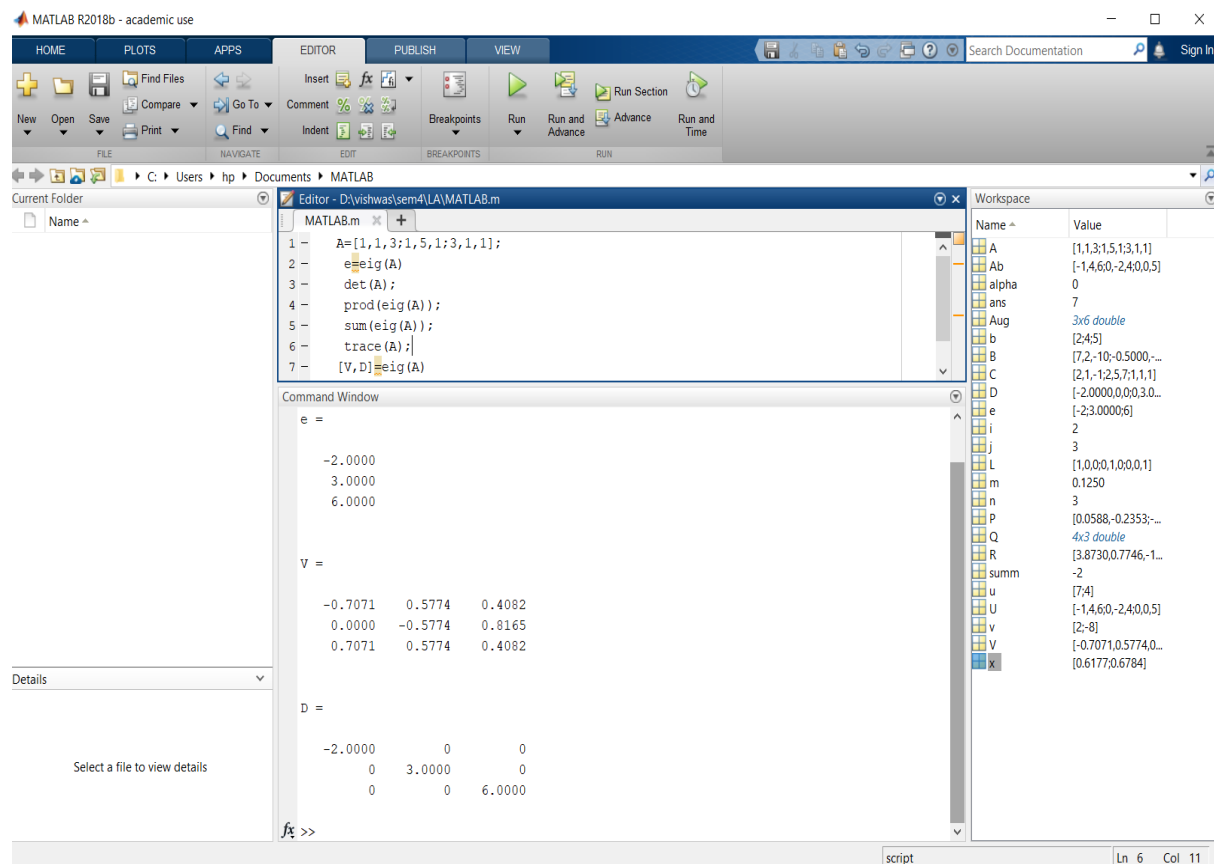
$\text{prod}(eig(A))$

$\text{sum}(eig(A))$

$\text{trace}(A)$

$[V,D]=eig(A)$

Output:

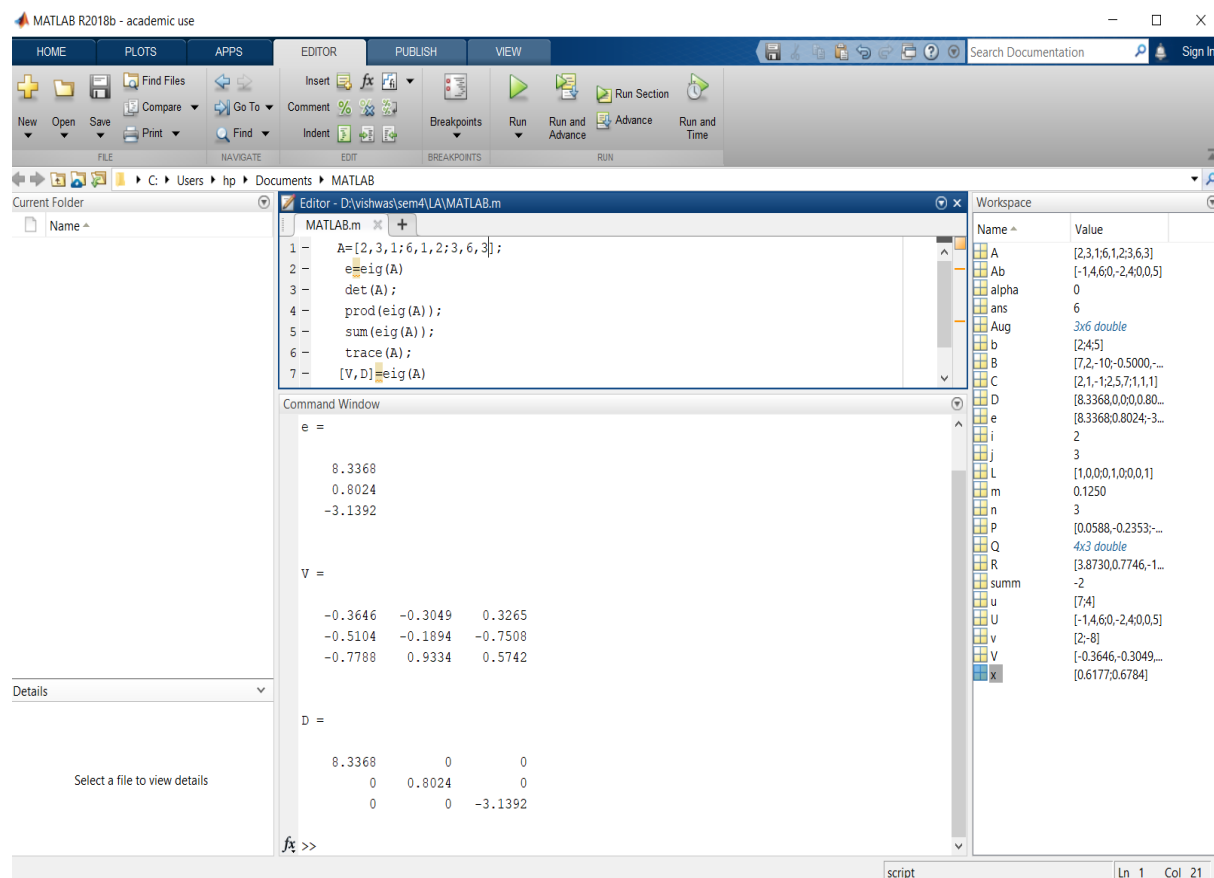


ii) Find the eigenvalues and the corresponding eigenvectors of the matrix

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 6 & 1 & 2 \\ 3 & 6 & 3 \end{bmatrix};$$

Ans: $A = \begin{bmatrix} 2 & 3 & 1 \\ 6 & 1 & 2 \\ 3 & 6 & 3 \end{bmatrix};$
 $e = \text{eig}(A)$
 $\det(A);$
 $\text{prod}(\text{eig}(A));$
 $\text{sum}(\text{eig}(A));$
 $\text{trace}(A);$
 $[V,D] = \text{eig}(A)$

Output:



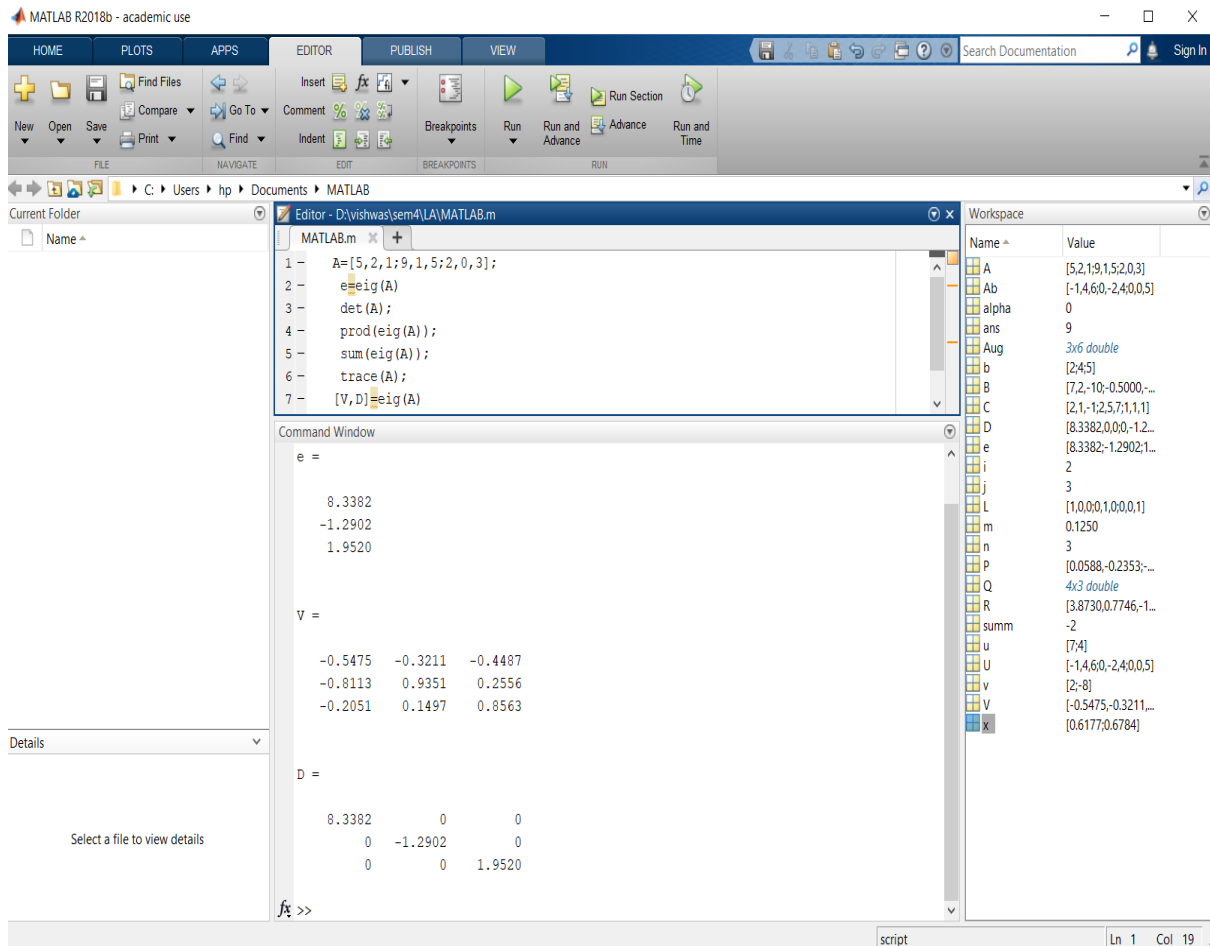
iii) Find the eigenvalues and the corresponding eigenvectors of the matrix

$$A = \begin{bmatrix} 5 & 2 & 1 \\ 9 & 1 & 5 \\ 2 & 0 & 3 \end{bmatrix};$$

Ans: $A = \begin{bmatrix} 5 & 2 & 1 \\ 9 & 1 & 5 \\ 2 & 0 & 3 \end{bmatrix};$
 $e = \text{eig}(A)$
 $\det(A);$
 $\text{prod}(\text{eig}(A));$
 $\text{sum}(\text{eig}(A));$

```
trace(A);
[V,D]=eig(A)
```

Output



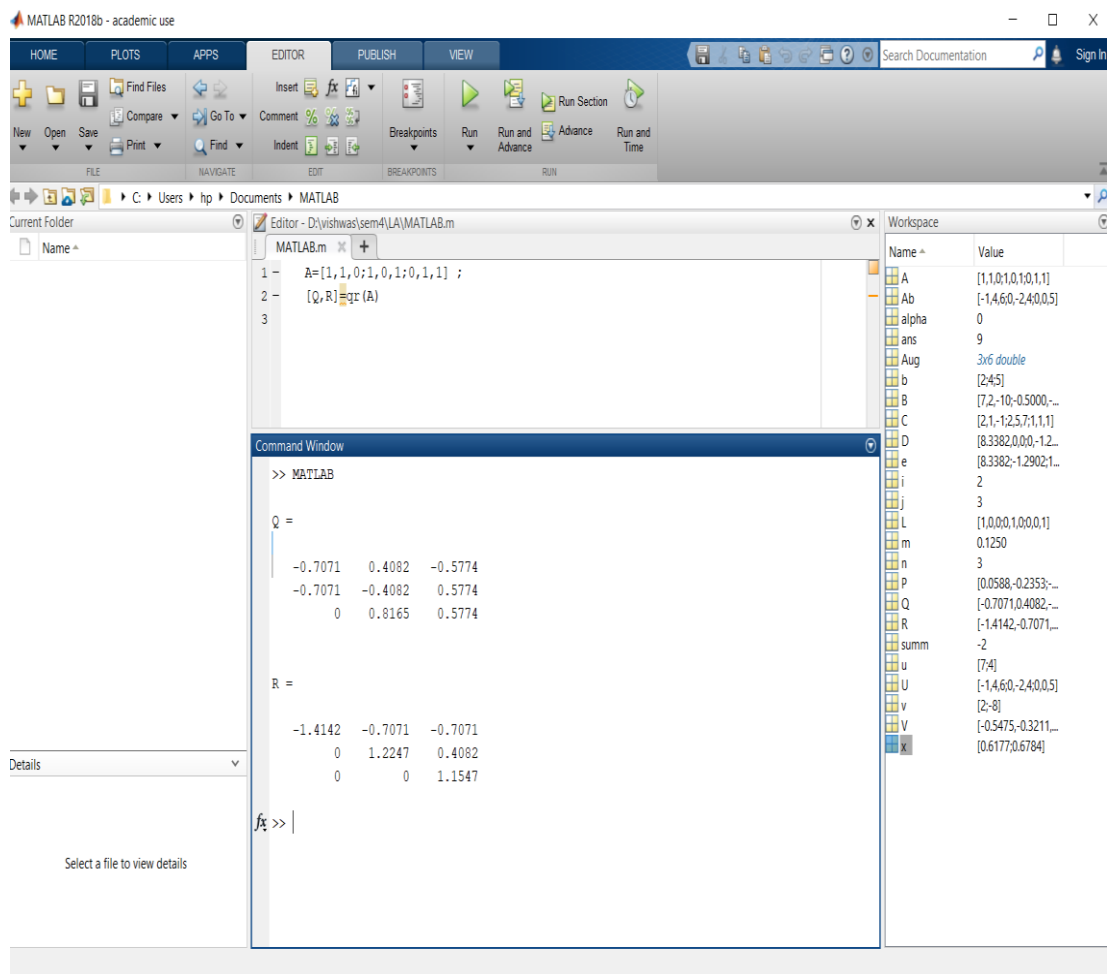
4) QR Decomposition with Gram-Schmidt:

i) Find QR factorization of the matrix $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

$A=[1,1,0;1,0,1;0,1,1]$

$[Q,R]=qr(A)$

Output:



ii) Find QR factorization of the matrix $A = \begin{bmatrix} 1 & -1 & 4 \\ 1 & 4 & -2 \end{bmatrix}$

$A = [1, -1, 4; 1, 4, -2]$

$[Q, R] = \text{qr}(A)$

Output:

The image shows the MATLAB R2018b interface. The Editor window displays the following code in MATLAB.m:

```
1 A = [1, -1, 4; 1, 4, -2];  
2 [Q, R] = qr(A);  
3
```

The Command Window shows the output of the code:

```
>> MATLAB  
  
A =  
  
     1     -1     4  
     1     4     -2  
  
Q =  
  
   -0.7071   -0.7071  
   -0.7071    0.7071  
  
R =  
  
   -1.4142   -2.1213   -1.4142  
         0    3.5355   -4.2426  
  
fx >>
```

The Workspace window shows the following variables:

Name	Value
A	[1, -1, 4; 1, 4, -2]
Ab	[-1, 4, 6; 0, -2, 4; 0, 0, 5]
alpha	0
i	3
L	[1, 0, 0; 1, 0, 0; 1, 1]
n	3
Q	[-0.7071, -0.7071; ...]
R	[-1.4142, -2.1213; ...]
U	[-1, 4, 6; 0, -2, 4; 0, 0, 5]

iii) Find QR factorization of the matrix $A = \begin{bmatrix} 3 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 3 \end{bmatrix}$

$A = [3, 2, 4; 2, 0, 2; 4, 2, 3]$

$[Q, R] = \text{qr}(A)$

Output:

MATLAB R2018b - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

Find Files Compare Go To Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Current Folder: C:\Users\hp\Documents\MATLAB

Editor - D:\vishwas\sem4\LA\MATLAB.m

```
1 A=[3,2,4;2,0,2;4,2,3];
2 [Q,R]=qr(A)
3
```

Command Window

```
>> MATLAB

Q =

-0.5571    0.4952   -0.6667
-0.3714   -0.8666   -0.3333
-0.7428    0.0619    0.6667

R =

-5.3852   -2.5997   -5.1995
     0     1.1142    0.4333
     0         0   -1.3333

fx >>
```

Workspace

Name	Value
A	[3,2,4;2,4,2,3]
Ab	[-1,4,6;0,-2,4,0,5]
alpha	0
ans	9
Aug	3x6 double
b	[2,4,5]
B	[7,2,-10;-0,5000,~...
C	[2,1,-12,5,7;1,1,1]
D	[8,3382,0,0,0,-1,2...
e	[8,3382,-1,2902,1...
i	2
j	3
L	[1,0,0,1,0,0,0,1]
m	0.1250
n	3
P	[0,0588,-0,2353;~...
Q	[-0,5571,0,4952,~...
R	[-5,3852,-2,5997,~...
summ	-2
u	[7,4]
U	[-1,4,6;0,-2,4,0,5]
v	[2;-8]
V	[-0,5475,-0,3211,~...
x	[0,6177;0,6784]

Details

Select a file to view details

script Ln 1 Col 23