VISUALISATION OF EIGEN VECTORS

AIM: To write MATLAB code for Visualization of Eigen vectors.

Syntax used:

Command	Description
clc	Clears the command window.
Clear all	Clears all the before data.
eig	eig(A) returns a column vector containing the eigen values of square matrix A.
linspace	To generate points.
hold on	Waits for execution.
pause	Stops execution for a while.
plot	Plots the values in a graph.

MATLAB CODE:

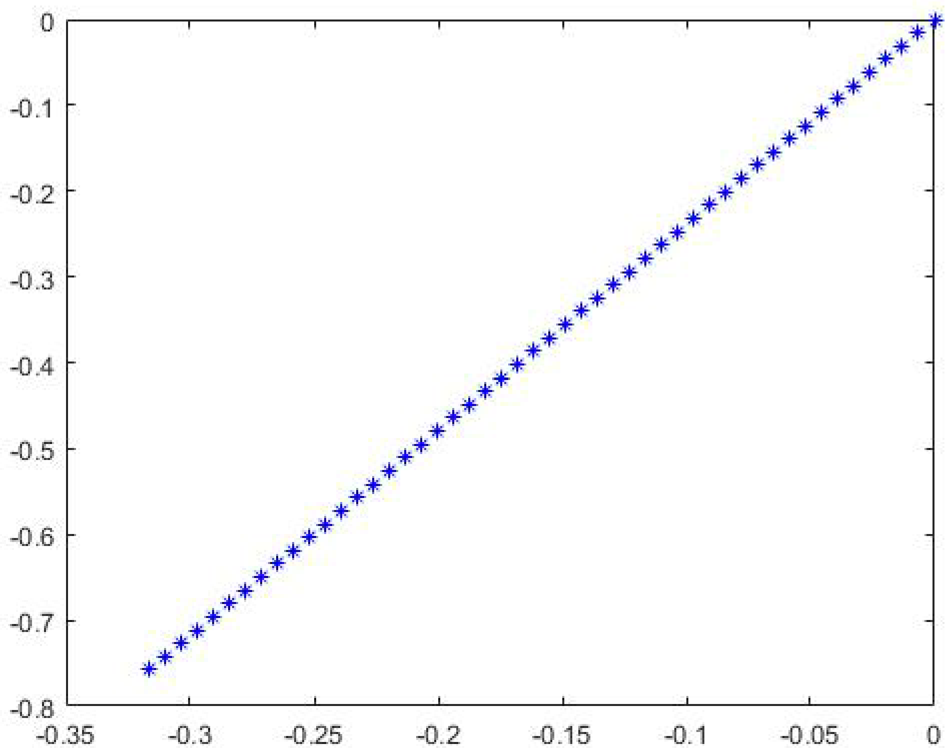
```
clc
clear all
A=input ('Enter a 2*2 matrix: ');
[p, d] =eig(A);
t=linspace (0,1,50);
plot (p (1,1) *t, p (2,1) *t,'*b')
hold on
pause
plot (d (1,1) *p (1,1) *t, d (1,1) *p (2,1) *t,'+y')
hold on
pause
plot (p (1,2) *t, p (2,2) *t,'*g')
```

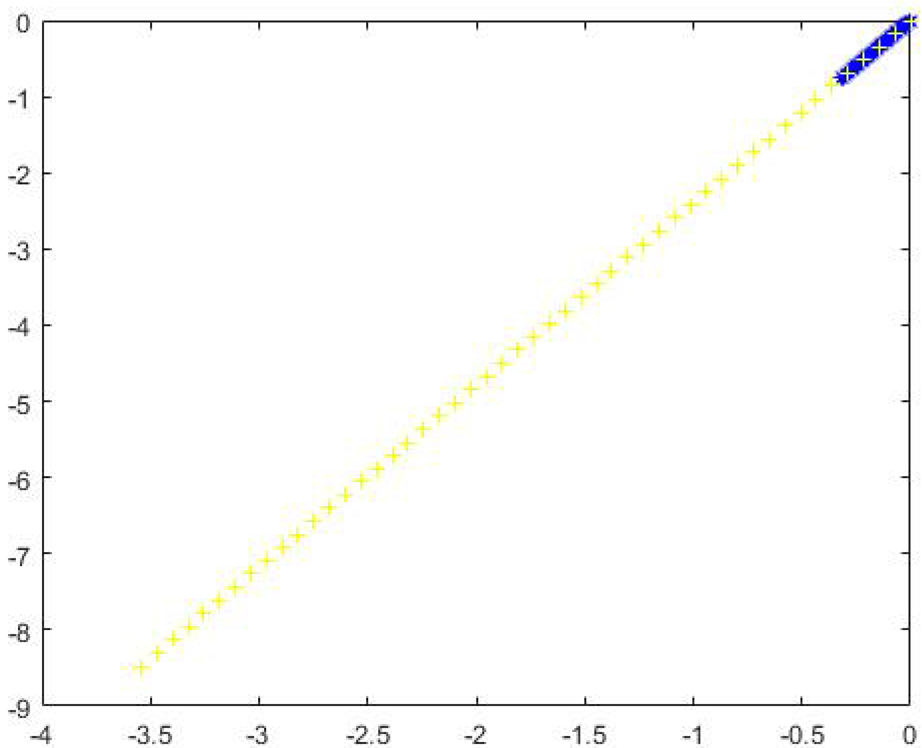
```
hold on
pause
plot (d (2,2) *p (1,2) *t, d (2,2) *p (2,2) *t,'+k')
```

EXERCISE:

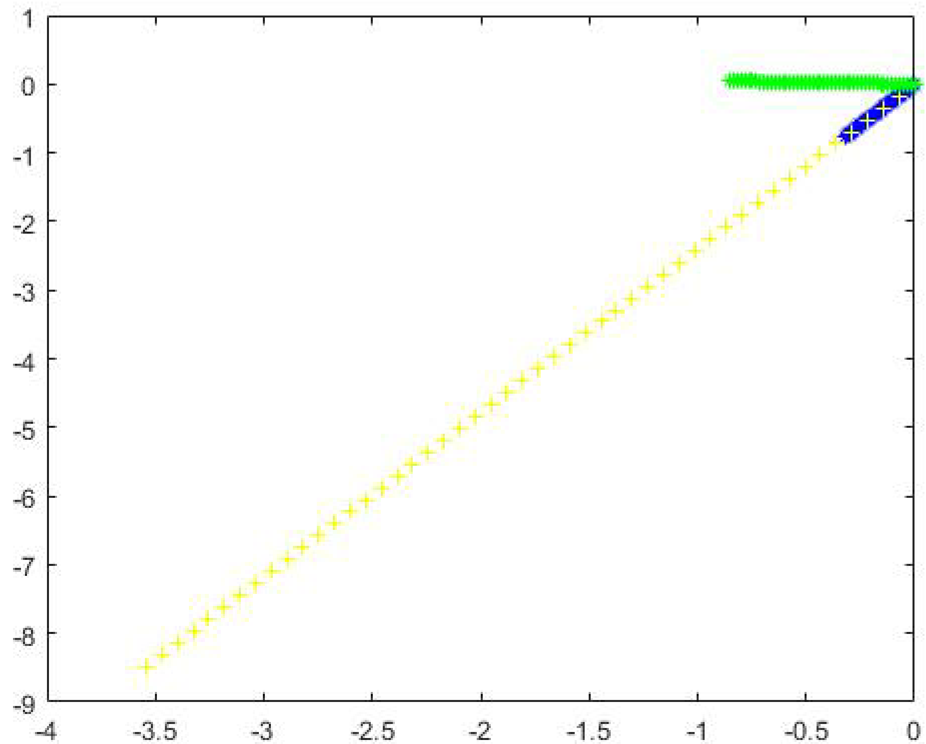
1. Write a MATLAB code to visualize the eigen value and eigen vector for the matrix $A=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 0\end{bmatrix}$.

OUTPUT:

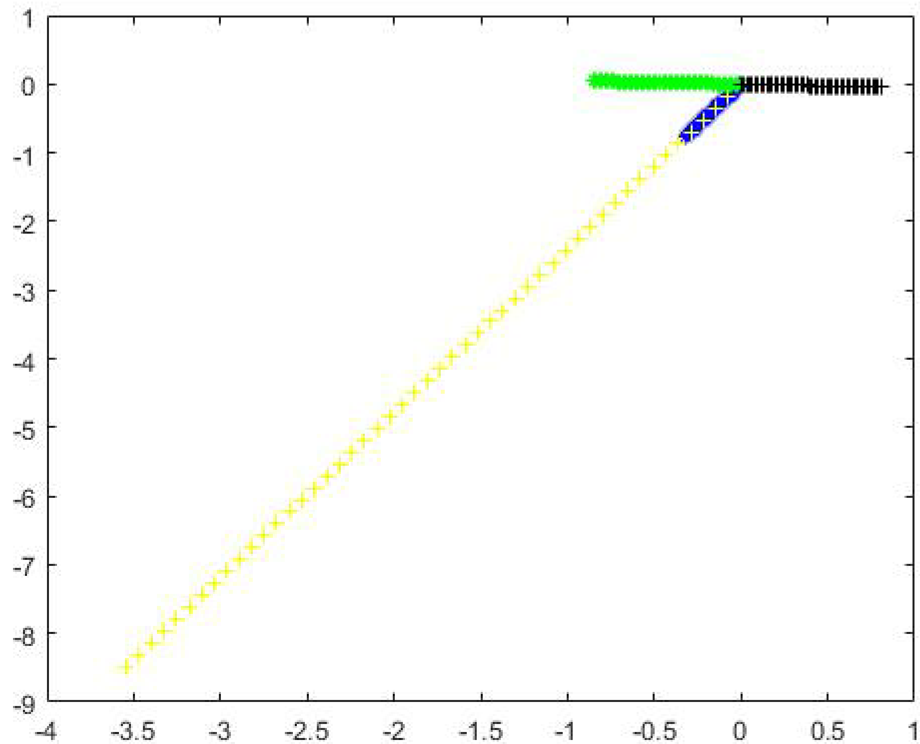




0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0



1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0


1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1

PROPERTIES OF EIGEN VECTORS

AIM: To show the properties of Eigen Vectors for a given Matrix.

Syntax Used:

Command	Description
poly	A is a Matrix. Poly returns the coefficients of polynomial of the Matrix A.
round	Rounds each root to the nearest integer. In this case of tie where element has fractional part exactly 0.5 round away from 0.
eig	eig(A) returns a column vector containing the eigen values of square matrix A.
[V, D]	D is eigen values and matrix V whose column are corresponded so that a*V=V*D.
inv	Used for finding the inverse of a matrix.
trace	finding sum of diagonal elements
det	Used to find the determinant of a given matrix.
transpose	Used for transpose of a matrix.

Matrix and its transpose have the same eigen value

Matlab code:

```
clc
clear all
A=[1 2 3;4 5 6;7 8 9]
e=eig(A)
[V ,E]=eig(A)
X=A'
r=eig(X)
[Q ,D]=eig(X)
```

Output:

```
A =

    1    2    3

    4    5    6

    7    8    9

e =

  16.1168

  -1.1168

  -0.0000

V =

  -0.2320  -0.7858   0.4082

  -0.5253  -0.0868  -0.8165
```

```
  -0.8187    0.6123    0.4082

E =

   16.1168         0         0

         0   -1.1168         0

         0         0   -0.0000


X =

     1     4     7

     2     5     8

     3     6     9

r =

   16.1168

   -1.1168

   -0.0000

Q =

 -0.4645   -0.8829    0.4082

  -0.5708   -0.2395   -0.8165

  -0.6770    0.4039    0.4082

D =

   16.1168         0         0

         0   -1.1168         0
```

```
         0         0   -0.0000
```

Eigenvalues of a triangular matrix are the diagonal elements of the matrix

Matlab code:

```
clc
clear all
A=[1 4 1;0 6 4;0 0 1]
e=eig(A)
```

output:

```
A =

     1     4     1

     0     6     4

     0     0     1

e =

    1

     6

     1
```

Product of eigen values is equal to the determinant of the matrix

Matlab code

```
clc
clear all
A=[2 1 3;3 5 4;7 9 5]
e=eig(A)
```

```
p=e(1,1)* e(2,1)* e(3,1)
det(A)
```

OUTPUT:

```
A =

    2    1    3

    3    5    4

    7    9    5

e =

  12.9609

   1.1860

  -2.1469

p =   -33.000

ans =-33
```

Sum of the eigenvalues is equal to the Trace of the matrix

Matlab code

```
clc
clear all
A=[1 2 3;4 5 6;7 8 9]
e=eig(A)
p=e(1,1)+e(2,1)+e(3,1)
t=A(1,1)+A(2,2)+A(3,3)
```

OUTPUT:

```
A =

    1    2    3

    4    5    6

    7    8    9

e =

  16.1168

  -1.1168

  -0.0000

p = 15.0000

t =15
```

For a real symmetric matrix, the eigenvectors of distinct eigen values are orthogonal

Matlab code:

```
clc
clear all
A=[1 0 0;0 2 0;0 0 3]
[V ,E]=eig(A)
u=[V(1,1),V(2,1),V(3,1)]
v=[V(1,2),V(2,2),V(3,2)]
w=[V(1,3),V(2,3),V(3,3)]
p=u*v'
q=v*w'
r=w*u'
```

OUTPUT:

```
A =

    1    0    0

    0    2    0

    0    0    3

V =

    1    0    0

    0    1    0

    0    0    1

E =

   1    0    0

    0    2    0

    0    0    3

u =  1    0    0

v =  0    1    0

w =  0    0    1

p = 0

q =  0

r = 0
```

Diagonalization

AIM: To write MATLAB code for Diagonalization of a Matrix.

Syntax used:

Command	Description
eig	eig(A) returns a column vector containing the eigen values of square matrix A.
[V, D]	D is eigen values and matrix V whose column are corresponded so that a*V=V*D.
inv	Used for finding the inverse of a matrix.
A’	Used to find the transpose of a Matrix.

MATLAB CODE:

```
clc
clear all
A=input ('Enter the matrix A: ');
N=length(A)
[X, D] =eig(A);
disp ('The eigen values of A are: ');
disp(diag(real(D)));
disp ('The eigen vectors for the corresponding eigen values');
disp(X)
%Diagonalization of the real matrix A
disp ('Modal matrix associated with A is ')
p=X
if A==A'
disp ('The given matrix is orthogonal')
```

```
D=p'*A*p
disp ('Thus A is reduce to the diagonal matrix D though p by orthogonal
transformation')
else
   D=inv(p)*A*p
   disp ('Thus A is reduced to the diagonal matrix D through p by similarity
transformation')
end
```

EXERCISE:

Write a MATLAB code for diagonalization using similarity transformation

$$A = \begin{bmatrix} -1 & 2 & -2 \\ 1 & 2 & 1 \\ -1 & -1 & 0 \end{bmatrix}.$$

OUTPUT:

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

N = 3

The eigen values of A are:
  -1.0000
   1.0000
   1.0000

The eigen vectors for the corresponding eigen values
  -0.8165 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0000i
   0.4082 + 0.0000i   0.7071 + 0.0000i   0.7071 + 0.0000i
  -0.4082 + 0.0000i  -0.7071 + 0.0000i  -0.7071 - 0.0000i

Modal matrix associated with A is

p =
  -0.8165 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0000i
   0.4082 + 0.0000i   0.7071 + 0.0000i   0.7071 + 0.0000i
```

-0.4082 + 0.0000i -0.7071 + 0.0000i -0.7071 - 0.0000i

D =
-1.0000 - 0.0000i -0.0000 + 0.0000i -0.0000 + 0.0000i
-0.0000 + 0.0000i   1.0000 + 0.0000i   0.0000 - 0.0000i
-0.0000 - 0.0000i   0.0000 + 0.0000i   1.0000 - 0.0000i

Thus A is reduced to the diagonal matrix D through p by similarity transformation

Write a MATLAB code for diagonalization using orthogonal transformation

$$A = \begin{bmatrix} 8 & -6 & 2 \\ -6 & 7 & -4 \\ 2 & -4 & 3 \end{bmatrix}.$$

OUTPUT:

Enter the matrix A: [8 -6 2; -6 7 -4; 2 -4 3]

N = 3

The eigen values of A are:
   0.0000
   3.0000
  15.0000

The eigen vectors for the corresponding eigen values
   0.3333   0.6667   -0.6667
   0.6667   0.3333   0.6667
   0.6667   -0.6667   -0.3333

Modal matrix associated with A is

p =
   0.3333   0.6667   -0.6667

```
    0.6667    0.3333    0.6667
    0.6667   -0.6667   -0.3333
```

The given matrix is orthogonal

```
D =
   -0.0000   -0.0000   -0.0000
   -0.0000    3.0000   -0.0000
   -0.0000   -0.0000   15.0000
```

Thus A is reduce to the diagonal matrix D though p by orthogonal transformation

Variation of parameters

AIM: To write the MATLAB code for Variation of parameters.

Syntax used:

Command	Description
solve	To solve an equation.
real	Real parts of roots.
imag	Imaginary part of roots.
exp	Exponential.
simplify	For simplifying equation.
diff	To differentiate equation.
int	To integrate an equation.

MATLAB CODE:

```
clc
clear all
syms m r c1 c2 x f2
a=input ('Enter the coefficient of D2y: ')
b=input ('Enter the coefficient of Dy: ')
c=input ('Enter the coefficient of y: ')
eq=a*m^2+b*m+c
r=solve(eq)
p=real (r (1))
```

```
q=imag (r (1))
if q~=0
   y1=exp(p*x) *cos(q*x)
   y2=exp(p*x) *sin(abs(q)*x)
else if r (1) ==r (2)
      y1=exp (r (1) *x)
      y2=x*exp (r (1) *x)
else
    y1=exp (r (1) *x)
    y2=exp (r (2) *x)
end
y_h=c1*y1+c2*y2
w=simplify(y1*diff(y2)-y2*diff(y1))
f=input ('Enter the non-homogeneous part: ')
y_p=-y1*int(y2*f/w) +y2*int(y1*f/w);
y1=simplify(y_h+y_p);
disp ('The general solution of the given ODE is ')
disp(y1)
```

EXERCISE:

$$y'' - 5y' + 6y = 2e^{x}$$

OUTPUT:

Enter the coefficient of D2y: 1

a = 1

Enter the coefficient of Dy: -5

b =-5

Enter the coefficient of y: 6

c = 6

eq =m^2 - 5*m + 6

r =2,3

p =2

q =0

y1 =exp(2*x)

y2 =exp(3*x)

y_h =c1*exp(2*x) + c2*exp(3*x)

w =exp(5*x)

Enter the non-homogeneous part: 2*exp(x)

f =2*exp(x)

The general solution of the given ODE is
exp(x)*(c1*exp(x) + c2*exp(2*x) + 1)

$$4y'' - 4y' + y = 16e^{\frac{x}{2}}$$

OUTPUT:

Enter the coefficient of D2y: 4

a = 4

Enter the coefficient of Dy: -4

b = -4

Enter the coefficient of y: 1

c = 1

eq =4*m^2 - 4*m + 1

r = 1/2,1/2

p =1/2

q =0

y1 =exp(x/2)

y2 =x*exp(x/2)

y_h =c1*exp(x/2) + c2*x*exp(x/2)

w =exp(x)

Enter the non-homogeneous part: 16*exp(x/2)

f =16*exp(x/2)

The general solution of the given ODE is
exp(x/2)*(8*x^2 + c2*x + c1)

Cayley Hamilton Theorem

AIM: To write MATLAB code for Cayley Hamilton Theorem.

Syntax used:

Command	Description
round	Used to round of the values to the nearest integer.
poly	Used for finding the coefficients of the polynomial.
inv	Used to find the inverse of a given Matrix.

MATLAB CODE:

```
clc
clear all
a= input ('Enter the Matrix A: ')
R= [1 0 0;0 1 0;0 0 1]
p= round(poly(a))
A= (a^2*p (1,1) +a*p (1,2) +R*p (1,3))/-p (1,4)
```

EXERCISE:

Write a MATLAB code to verify the Cayley – Hamilton theorem for the given matrix $A = \begin{bmatrix} 13 & -3 & 5 \\ 0 & 4 & 0 \\ -15 & 9 & -7 \end{bmatrix}$.

OUTPUT:

```
Enter the Matrix A:[13 -3 5;0 4 0;-15 9 7]

a =
   13   -3    5
    0    4    0
  -15    9    7

R =
    1    0    0
    0    1    0
    0    0    1


p = 1   -24   246  -664

A =

   0.0422    0.0994   -0.0301
        0    0.2500         0
   0.0904   -0.1084    0.0783
```

Cauchy Euler

AIM: To write MATLAB code for Cauchy Euler.

Syntax used:

Command	Description
clc	Clears the command window.
Clear all	Clears all the before data.
solve	To solve an equation.
real	Real parts of roots.
imag	Imaginary part of roots.
exp	Exponential.
simplify	For simplifying equation.
diff	To differentiate equation.
int	To integrate an equation.

MATLAB CODE:

```
clc
clear all
```

```
syms x k1 k2
a=input('enter the coefficient of D2y:');
b=input('enter the coefficient of Dy:');
c=input('enter the  coefficient of y:');
a1=a/a; b1=b/a; c1=c/a;
eq1=a1*x^2+b1*x+c1;
r=solve(eq1,'x')
if imag(r)~=0
   y1=exp(real(r(1))*x)*cos(imag(r(1))*x);
   y2=exp(real(r(1))*x)*sin(abs(imag(r(1)))*x);
elseif r(1)==r(2)
   y1=exp(r(1)*x)
   y2=x*exp(r(1)*x)
else
   y1=exp(r(1)*x)
   y2=exp(r(2)*x)
end
w=simplify(y1*diff(y2)-y2*diff(y1))
if w~=0
   y_c=k1*y1+k2*y2
   g=input('enter the non homogeneous part:')
   y_p=-y1*int(y2*g/w)+y2*int(y1*g/w)
   y_g=y_c+y_p
   disp(y_g)
end
```

EXERCISE:

$$x^2\frac{d^2y}{dx^2}-4x\frac{dy}{dx}+6y=\sin(\log x)$$

OUTPUT:

```
enter the coefficient of D2y:1
enter the coefficient of Dy:-4
```

enter the coefficient of y:6
r =
 2 - 2^(1/2)*1i
 2 + 2^(1/2)*1i

 w =
2^(1/2)*exp(4*x)

y_c =

k1*exp(2*x)*cos(2^(1/2)*x) + k2*exp(2*x)*sin(2^(1/2)*x)

enter the non homogeneous part:sin(log(x))

g =
 sin(log(x))

 y_p =

 exp(2*x)*sin(2^(1/2)*x)*int((2^(1/2)*sin(log(x))*exp(-2*x)*cos(2^(1/2)*x))/2, x) - exp(2*x)*cos(2^(1/2)*x)*int((2^(1/2)*sin(log(x))*exp(-2*x)*sin(2^(1/2)*x))/2, x)

 y_g =

k1*exp(2*x)*cos(2^(1/2)*x) + k2*exp(2*x)*sin(2^(1/2)*x) - (2^(1/2)*exp(2*x)*cos(2^(1/2)*x)*int(sin(log(x))*exp(-2*x)*sin(2^(1/2)*x), x))/2 + (2^(1/2)*exp(2*x)*sin(2^(1/2)*x)*int(sin(log(x))*exp(-2*x)*cos(2^(1/2)*x), x))/2

Cauchy legendary

<u>AIM:</u> To write MATLAB code for Cauchy legendary.

<u>Syntax used:</u>

Command	**Description**
clc	Clears the command window.
Clear all	Clears all the before data.
solve	To solve an equation.
real	Real parts of roots.
imag	Imaginary part of roots.
exp	Exponential.
simplify	For simplifying equation.
diff	To differentiate equation.
int	To integrate an equation.

<u>MATLAB CODE</u>

```
clc
clear all
syms m c1 c2 x t
a=input('enter a:')
b=input('enter b:')
p1=input('Enter the coefficient of (ax+b)^2*D2y:')
p2=input('Enter the coefficient of (ax+b)*D1y:')
p3=input('Enter the coefficient of y:');
f1=input('non homogeneous');
f=simplify(subs(f1,x,(exp(t)-b)/a))
eq=(a^2)*p1*m^2-(a^2)*p1*m+a*(p2)*m+p3
r=solve(eq,m)
p=real(r(1))
q=imag(r(1))
if q~=0
   y1=exp(p*t)*cos(q*t)
   y2=exp(p*t)*sin(abs(q)*t)
elseif r(1)==r(2)
   y1=exp(r(1)*t)
   y2=x*exp(r(1)*t)
else
   y1=exp(r(1)*t)
   y2=exp(r(2)*t)
end

y_h=c1*y1+c2*y2
w=simplify(y1*diff(y2)+y2*diff(y1));

y_p=-y1*int(y2*f/(a^2*p1*w),t)+y2*int(y1*f/(a^2*p1*w),t);
yy=(y_h+y_p);
y1=subs(yy,t,log(a*x+b))
disp('The general solution of the given ODE is')
disp(simplify(y1))
```

EXERCISE:

$$(2x+3)^2 y'' - (2x+3) y' + 2y = 6x$$

OUTPUT:

enter a:2

a =2

enter b:3
b =3

Enter the coefficient of (ax+b)^2*D2y:1

p1 =1

Enter the coefficient of (ax+b)*D1y:-1

p2 =-1

Enter the coefficient of y:2

non-homogeneous 6*x

f =3*exp(t) - 9

eq =4*m^2 - 6*m + 2

r =1/2,1

p =1/2

q =0

y1 =exp(t/2)

y2 = exp(t)

y_h =c2*exp(t) + c1*exp(t/2)

y1 =
c1*(2*x + 3)^(1/2) - 2*x + (2*x + 3)*(log(2*x + 3)/2 + 3/(2*(2*x + 3))) + c2*(2*x + 3) - 6

The general solution of the given ODE is
(log(2*x + 3)*(2*x + 3))/2 - 2*x + c1*(2*x + 3)^(1/2) + c2*(2*x + 3) - 9/2