**APPLIED MATHEMATICS-4 LAB**

***ETCS-211***

**Faculty Name:**

BHRAMA PRAKASH

**Student Name: Naman Munjal**

**Roll No.: 08514803119**

**Semester: Forth**

**Batch: IT**



**Department Of Information Technology**

Maharaja Agrasen Institute Of Technology, PSP area,

Sector-22, Rohini, New Delhi -110085

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | EXPERIMENT NAME | DATE | REMARKS |
| 1(A). | Write a program to perform addition of two matrices |  |  |
| 1(B). | Write a program to perform multiplication of two matrices |  |  |
| 1(C). | Write a program to find transpose of a matrix. |  |  |
| 2. | Write a program to find inverse of a matrix using Gauss Jordan Method |  |  |
| 3. | Write a program to find Eigen Values and Eigen Vectors of a given 2\*2 matrix |  |  |
| 4. | Write a program to find mean, standard deviation and first r moments about mean of given grouped data |  |  |
| 5(A). | To fit a straight line for given n pairs of values (x,y) |  |  |
| 5(B). | To fit a straight line for given n pairs of values (x,y) |  |  |
| 6(A). | Write a program to plot unit step function |  |  |
| 6(B). | Write a program to plot a square wave function |  |  |
| 7(A). | Write a program to find solution of a non-linear equation using Bisection method |  |  |
| 7(B). | Write a program to find solution of an non-linear equation using Newton - Raphson method |  |  |
| 8(A). | Write a program to evaluate a definite integral using Trapezoidal rule |  |  |
| 8(B). | Write a program to evaluate a definite integral using Simpson’s one third rule |  |  |
| 8(C). | Write a program to evaluate a definite integral using Simpson’s three eighth rule |  |  |
| 9. | Write a program to find the initial value problem using Euler’s method |  |  |
| 10. | Write a program to find the solution of initial value using Runge-Kutta method of fourth order |  |  |

**Experiment – 1a**

**Aim –** Write a program to perform addition of two matrices

//matrixaddition

m=input('Enter The Number Of Rows:')

n=input('Enter The Number Of Columns:')

A=zeros(m,n)

B=zeros(m,n)

C=zeros(m,n)

disp('Enter the elements of first matrix row wise:')

for i=1:m

for j=1:n

A(i,j)=input('')

end

end

disp('Enter the elements of second matrix row wise')

for i=1:m

for j=1:n

B(i,j)=input('')

end

end

for i=1:m

for j=1:n

C(i,j)=A(i,j)+B(i,j)

end

end

disp('First Matrix==>')

disp(A)

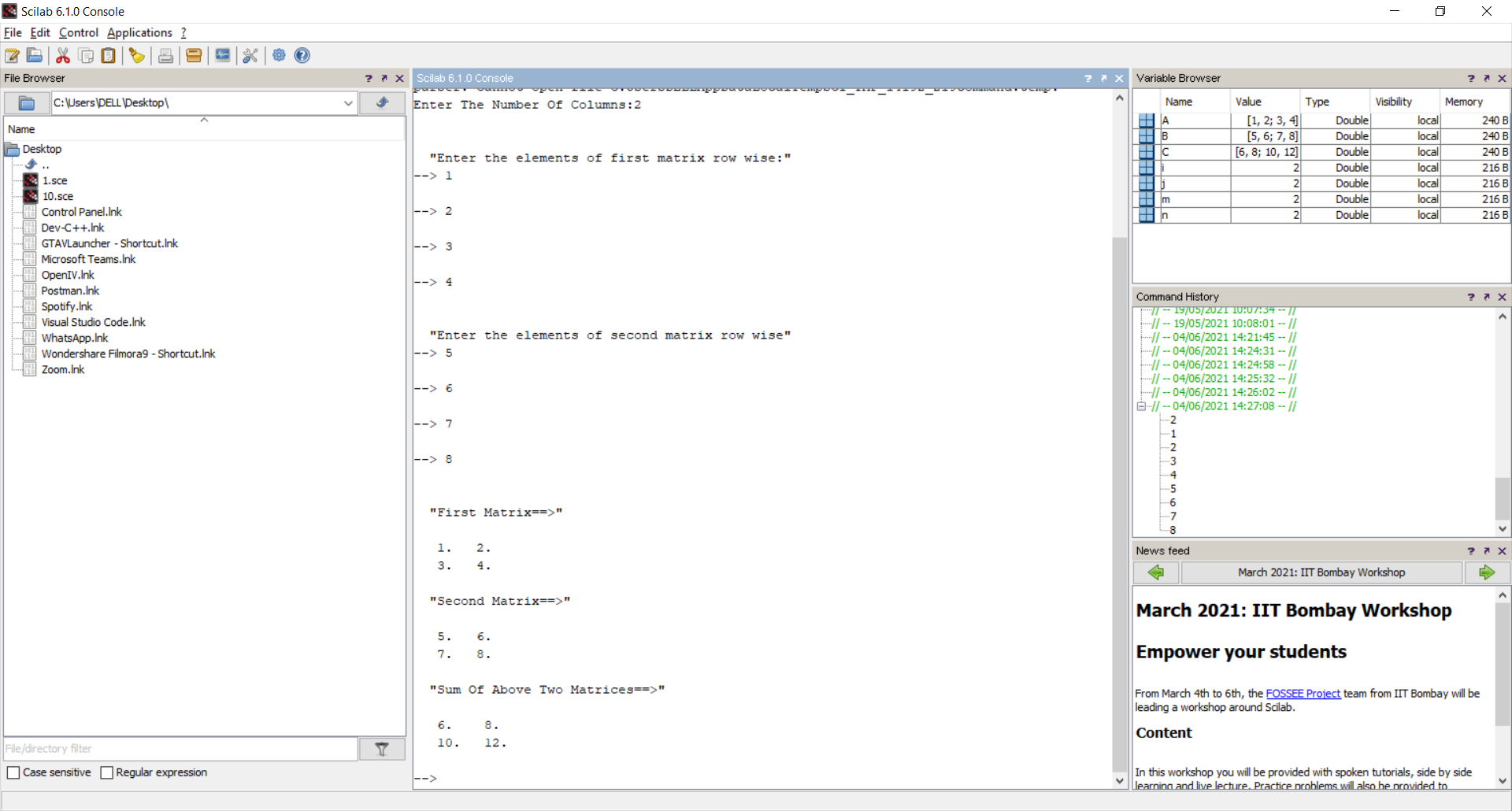
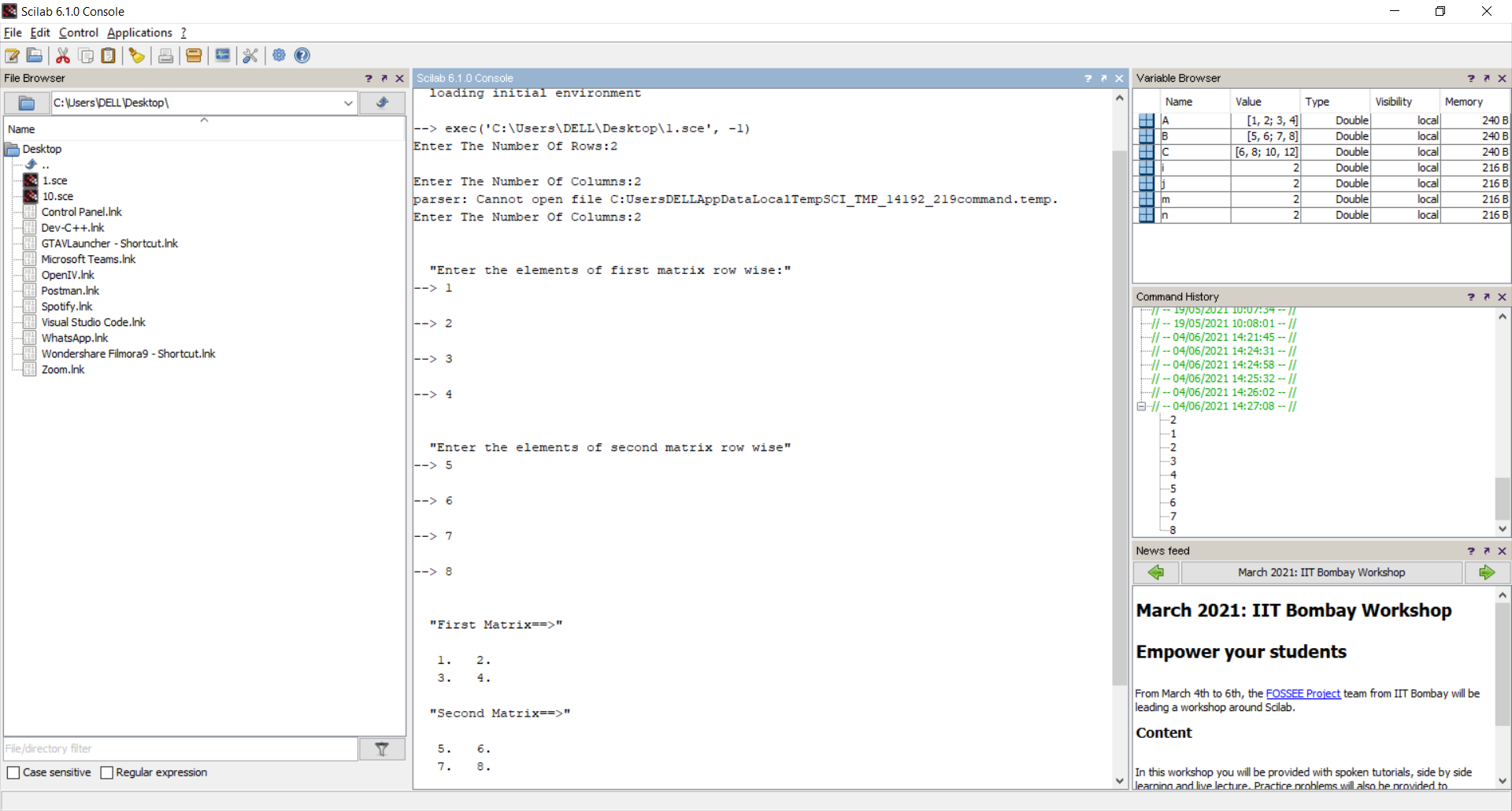
disp('Second Matrix==>')

disp(B)

disp('Sum Of Above Two Matrices==>')

disp(C)

OUTPUT:-



**Experiment – 1b**

**Aim –** Write a program to perform multiplication of two matrices

//matrix multiplication

m=input('Enter The Number Of Rows:')

n=input('Enter The Number Of Columns:')

A=zeros(m,n)

B=zeros(m,n)

C=zeros(m,n)

disp('Enter the elements of first matrix row wise:')

for i=1:m

for j=1:n

A(i,j)=input('')

end

end

disp('Enter the elements of second matrix row wise')

for i=1:m

for j=1:n

B(i,j)=input('')

end

end

C=A\*B

disp('First Matrix==>')

disp(A)

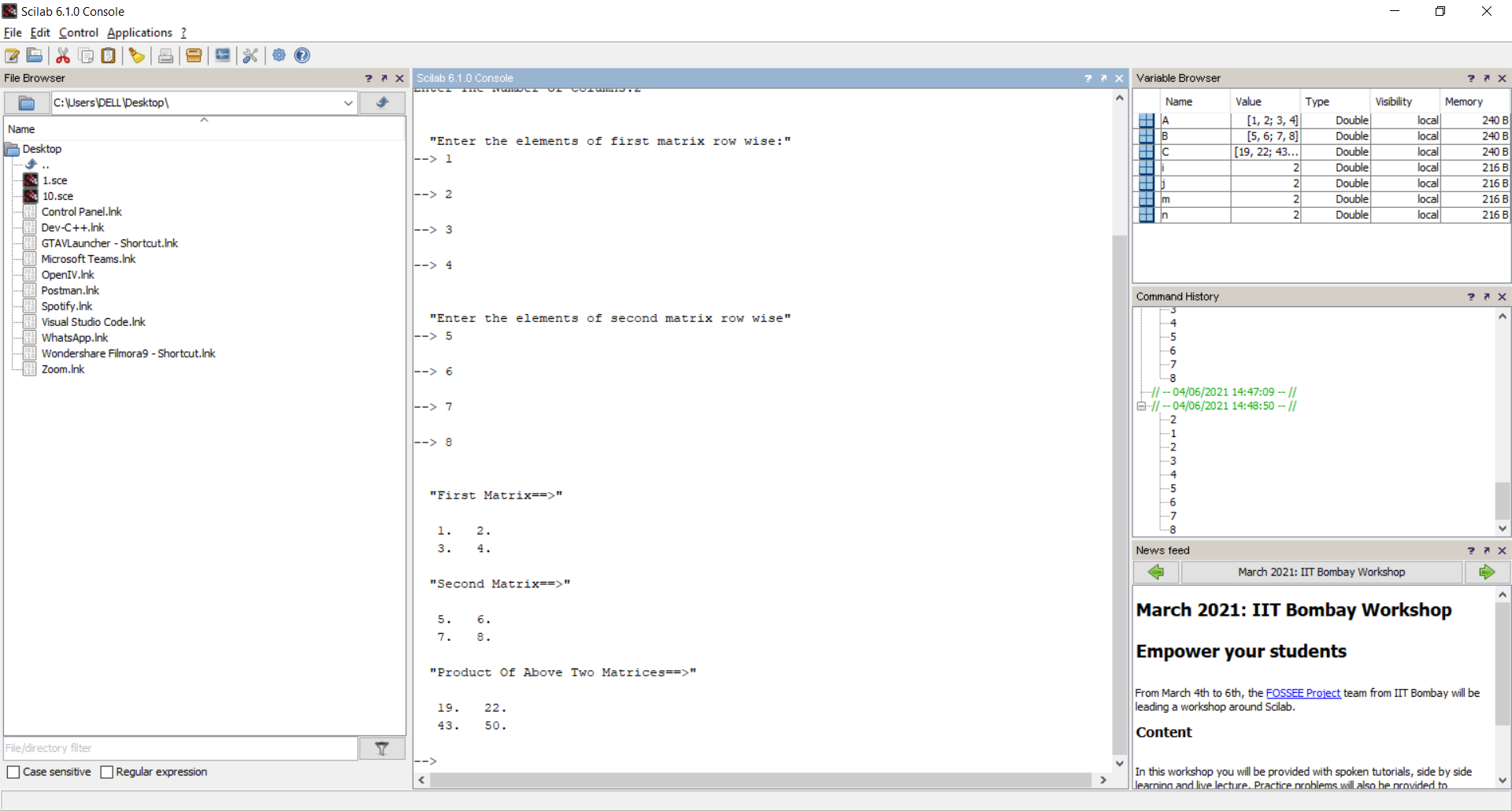
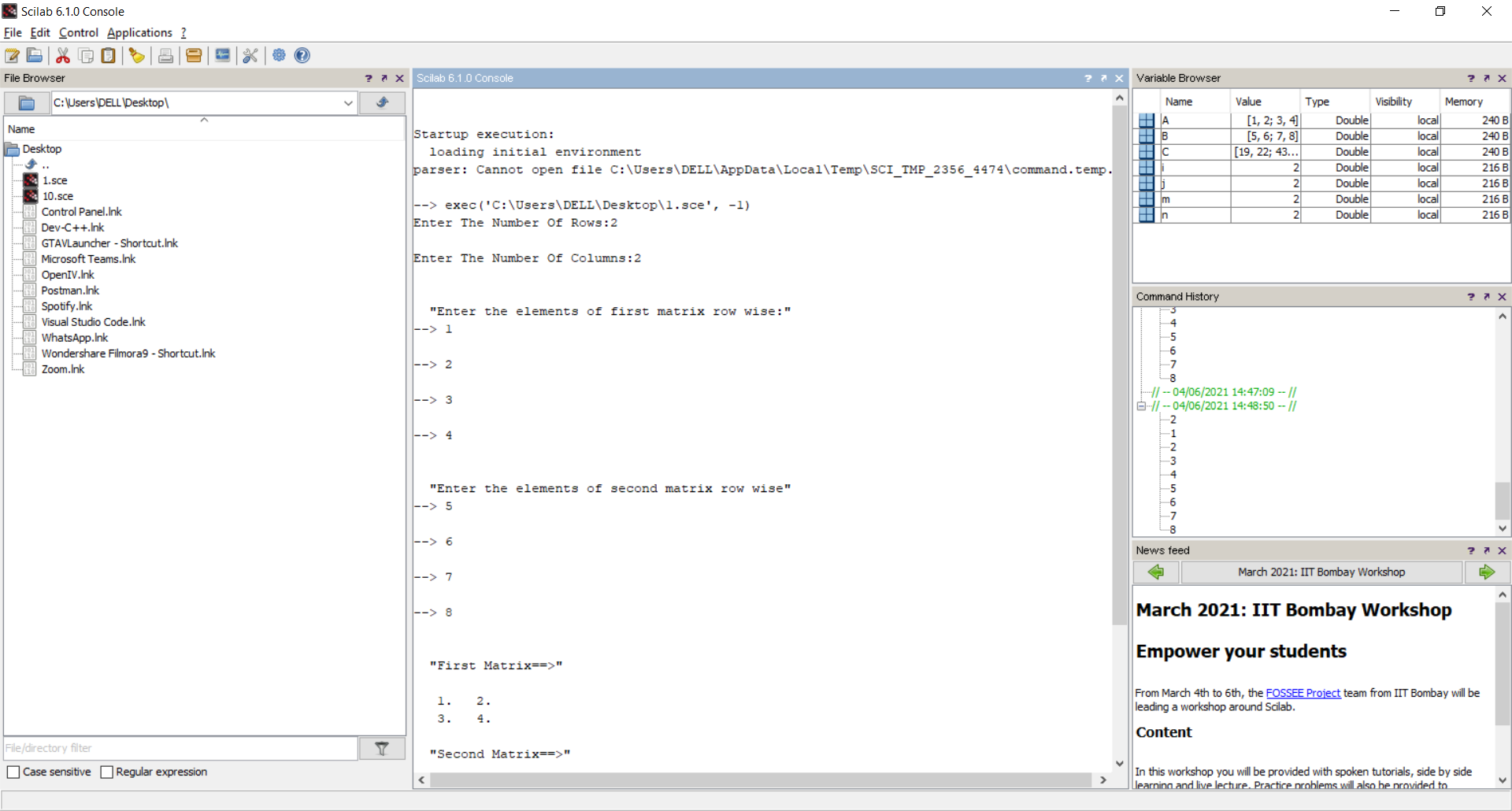
disp('Second Matrix==>')

disp(B)

disp('Product Of Above Two Matrices==>')

disp(C)

OUTPUT:-



**Experiment – 1c**

**Aim –** Write a program to find transpose of a matrix

//matrix transpose

m=input('Enter The Number Of Rows:')

n=input('Enter The Number Of Columns:')

A=zeros(m,n)

B=zeros(m,n)

disp('Enter the elements of first matrix row wise:')

for i=1:m

for j=1:n

A(i,j)=input('')

end

end

B=A'

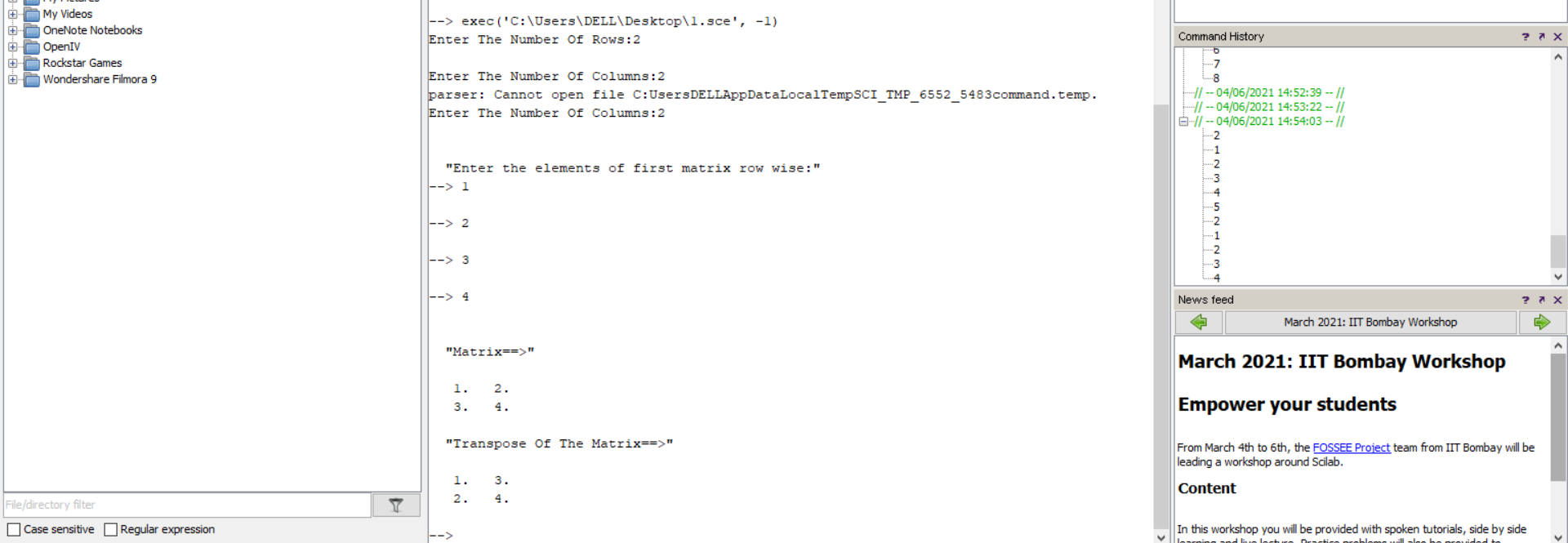
disp('Matrix==>')

disp(A)

disp('Transpose Of The Matrix==>')

disp(B)

OUTPUT:-



**Experiment – 2**

**Aim –** Write a program to find inverse of a matrix using Gauss Jordan Method

//Program to find inverse of any 3 by 3 matrix by Gauss Jordan Method

clc;

clear;

close;

function[B]=matinv(A)

    B=eye(3,3);

    disp('The given matrix A is==>')

    disp(A)

    if(det(A)==0) then

        disp('Matrix A is singular, Inverse does not exist')

        abort

    end

    aug=[A,B];

    if(aug(1,1)==0 &aug(2,1)~=0) then

        C(1,:)=aug(1,:);

        aug(1,:)=aug(2,:);

        aug(2,:)=C(1,:);

    elseif(aug(1,1)==0 &aug(3,1)~=0)

        C(1,:)=aug(1,:);

        aug(1,:)=aug(3,:);

        aug(3,:)=C(1,:);

    end

    aug(1,:)=aug(1,:)/aug(1,1);

    aug(2,:)=aug(2,:)-aug(2,1)\*aug(1,:);

    aug(3,:)=aug(3,:)-aug(3,1)\*aug(1,:);

    if(aug(2,2)==0)

        C(2,:)=aug(2,:);

        aug(2,:)=aug(3,:);

        aug(3,:)=C(2,:);

    end

    aug(2,:)=aug(2,:)/aug(2,2);

    aug(1,:)=aug(1,:)-aug(1,2)\*aug(2,:);

    aug(3,:)=aug(3,:)-aug(3,2)\*aug(2,:);

    aug(3,:)=aug(3,:)/aug(3,3);

    aug(1,:)=aug(1,:)-aug(1,3)\*aug(3,:);

    aug(2,:)=aug(2,:)-aug(2,3)\*aug(3,:);

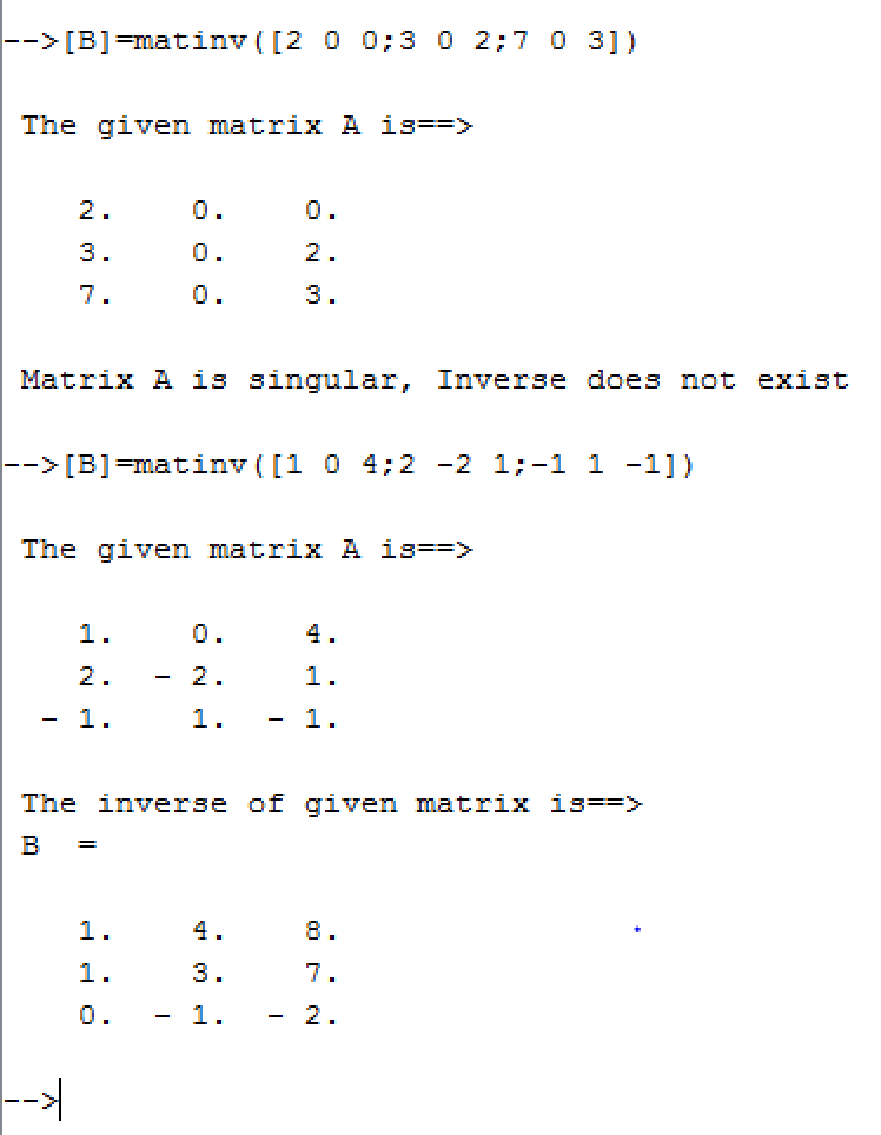
    aug(:,1:3)=[]

    B=aug(:,1:3);

    disp('The inverse of given matrix is==>')

endfunction

OUTPUT:-



**Experiment – 3**

**Aim –** Write a program to find Eigen Values and Eigen Vectors of a given 2\*2 matrix

*//Eigen values and Eigen vectors*

clc;

disp('Enter the matrix:')

for i=1:2

for j=1:2

A(i,j)=input('\');

end

end

b=A(1,1)+A(2,2);

c=A(1,1)\*A(2,2)-A(1,2)\*A(2,1);

disp('The characteristic equation is:')

disp(['e^2+' string(-b) '\*e+' string(c) '=0'])

e1=(b+sqrt(b^2-4\*c))/2;

e2=(b-sqrt(b^2-4\*c))/2;

if A(1,2)~=0 then

    v1=[A(1,2);e1-A(1,1)];

    v2=[A(1,2);e2-A(1,1)];

else if A(2,1)~=0

        v1=[e1-A(2,2);A(2,1)];

        v2=[e2-A(1,2);A(2,1)];

else

         v1=[1;0];

         v2=[0;1];

end

end

disp('First Eigen value is:');

disp(e1);

disp('First Eigen vector is:');

disp(v1);

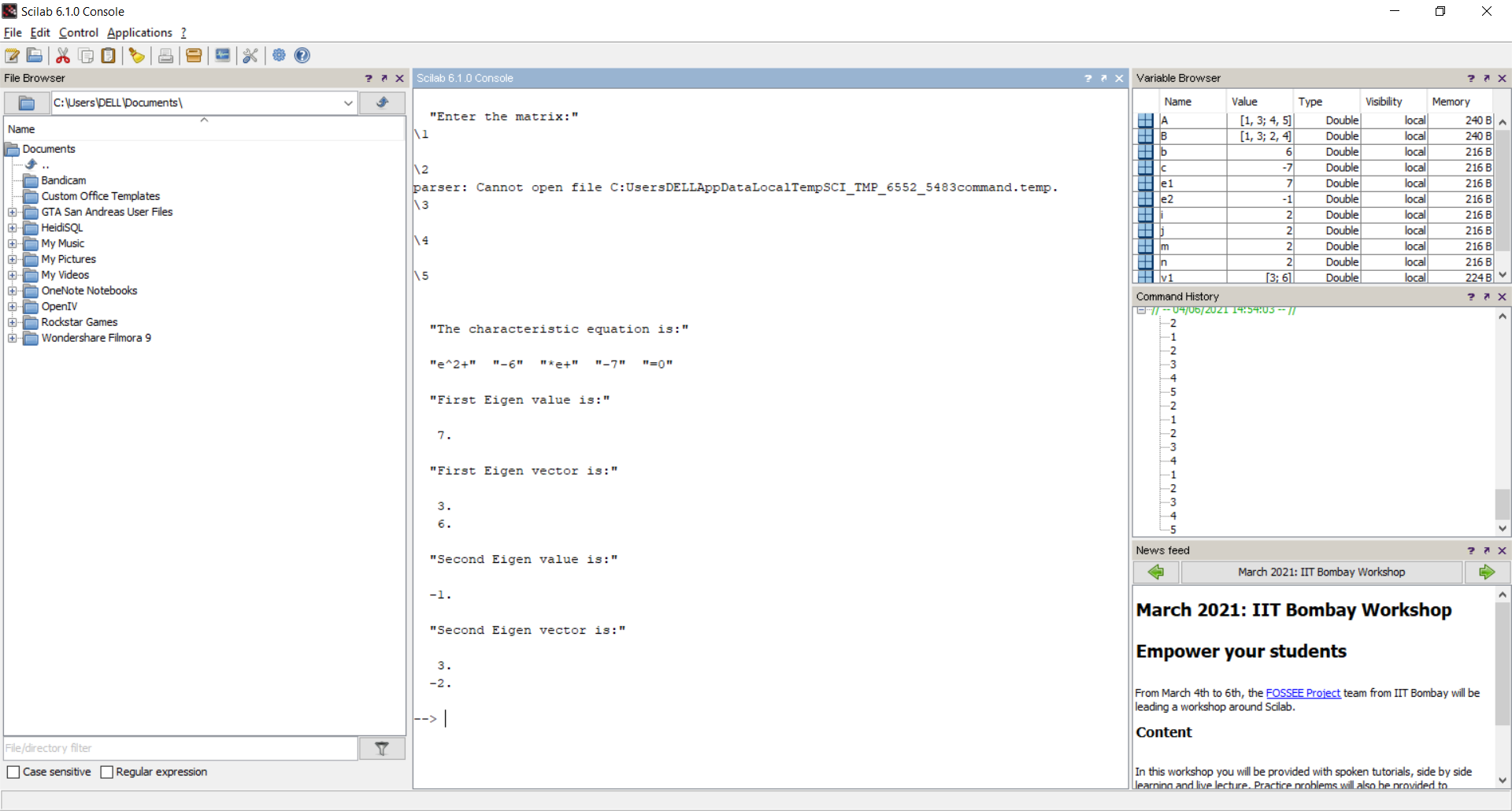
disp('Second Eigen value is:');

disp(e2);

disp('Second Eigen vector is:');

disp(v2);

OUTPUT:-



**Experiment – 4**

**Aim –** Write a program to find mean, standard deviation and first r moments about mean of given grouped data.

*//Program to find mean, standard deviation and first r moments about mean of*

*//given grouped data*

clc;

n=input('Enter the number of observations:');

disp('Enter the values of xi==>');

for i=1:n

    x(i)=input('\');

end;

disp('Enter thecorresponding frequencies fi==>');

sum3=0;

for i=1:n

    f(i)=input('\');

    sum3=sum3+f(i);

end;

r=input('Enter the number of moments to be caculated:');

sum1=0;

for i=1:n

    sum1=sum1+f(i)\*x(i);

end;

A=sum1/sum3;      *//calculation of mean*

printf('Mean=%f\n',A);

for j=1:r

    sum2=0;

    for i=1:n

        y(i)=f(i)\*(x(i)-A)^j;

        sum2=sum2+y(i);

    end;

    M(j)=(sum2/sum3);       *//calculation of moments*

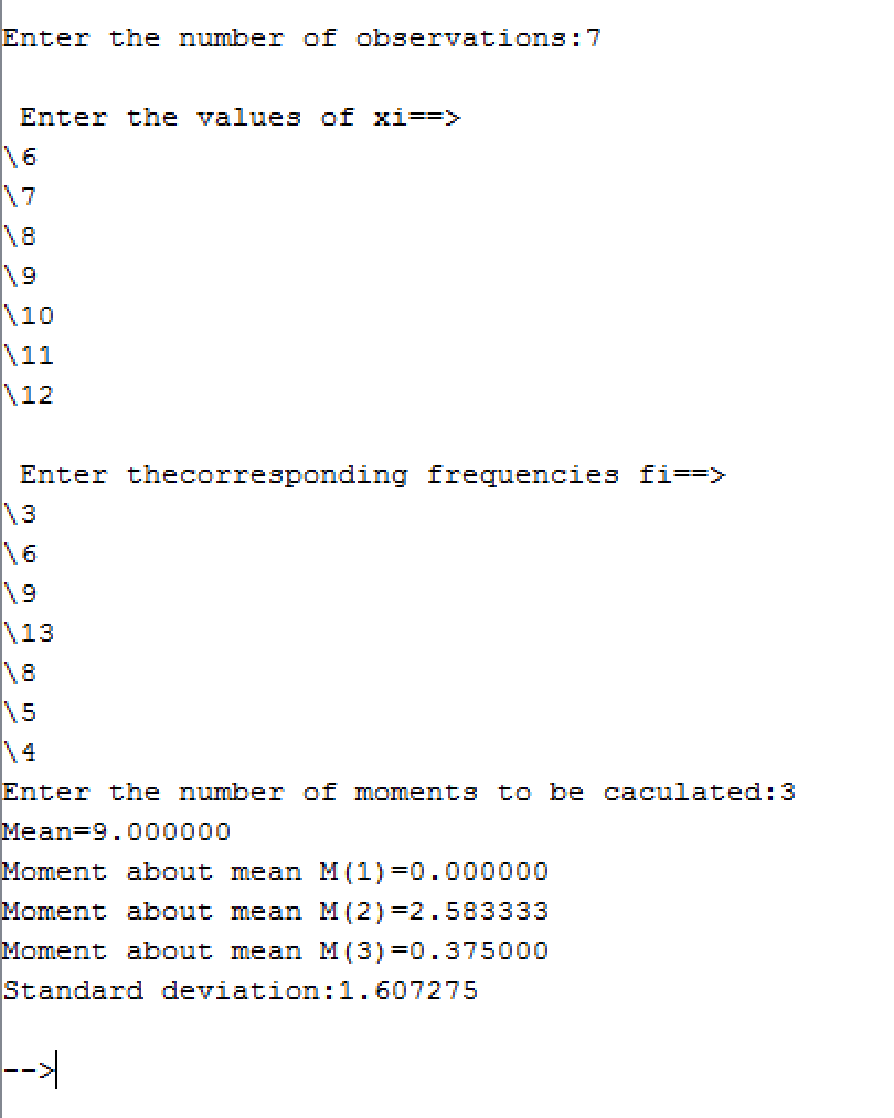
printf('Moment about mean M(%d)=%f\n',j,M(j));

end;

sd=sqrt(M(2));        *//calculation of standard deviation*

printf('Standard deviation:%f\n',sd);

OUTPUT:-



**Experiment – 5a**

**Aim –** To fit a straight line for given n pairs of values (x,y)

*//Program of straight line fitting for n given pairs of values*

clc;

clear;

close;

n=input('Enter the number of pairs of input(x,y):');

s1=0;

s2=0;

s3=0;

s4=0;

disp('Enter the values of x:')

for i=1:n

x(i)=input('\');

    s1=s1+x(i);

    s2=s2+x(i)\*x(i);

end

disp('Enter coressponding values of y:')

for i=1:n

y(i)=input('\');

    s3=s3+y(i);

end

for i=1:n

    s4=s4+x(i)\*y(i);

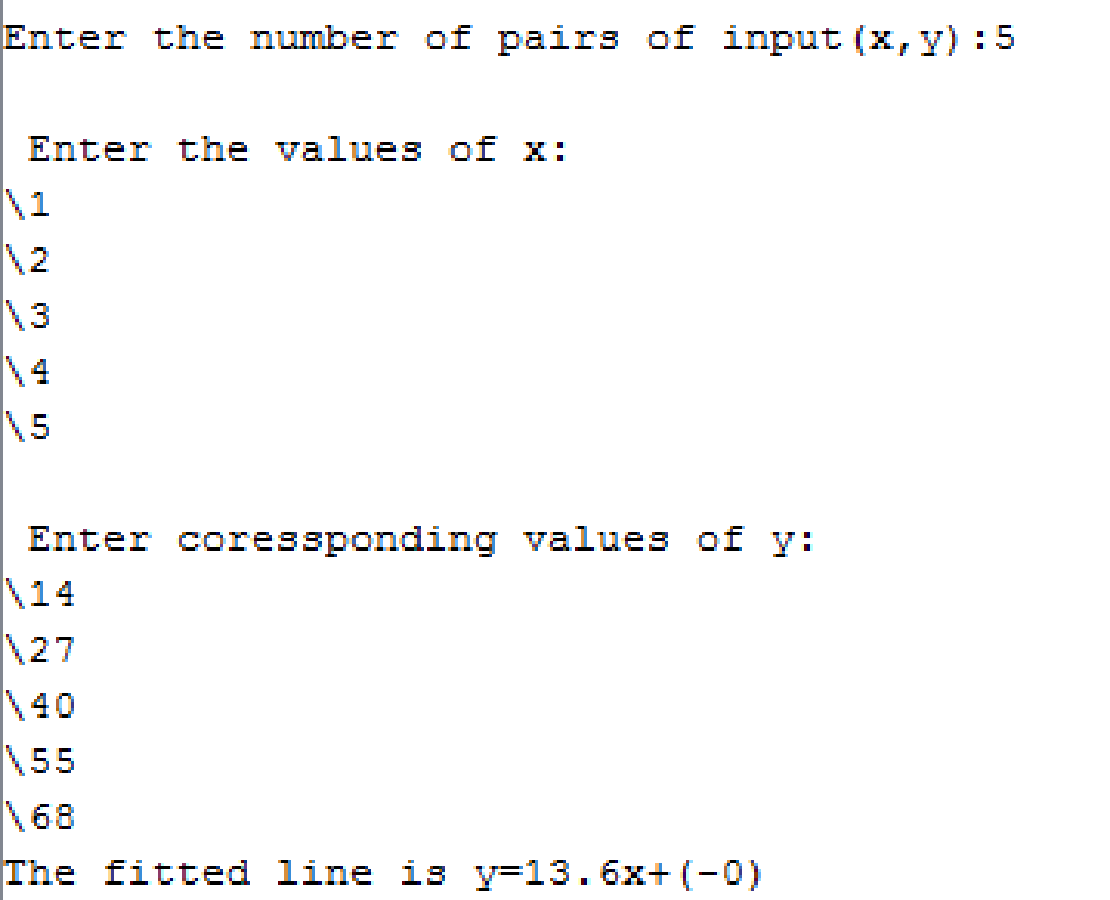
end

m=(s1\*s3-n\*s4)/(s1\*s1-n\*s2);

c=(s1\*s4-s2\*s3)/(s1\*s1-n\*s2);

printf('The fitted line is y=%gx+(%g)',m,c)

OUTPUT:-



**Experiment – 6a**

**Aim –** Write a program to plot unit step function

*//Program to plot unit step function*

clc;

clear;

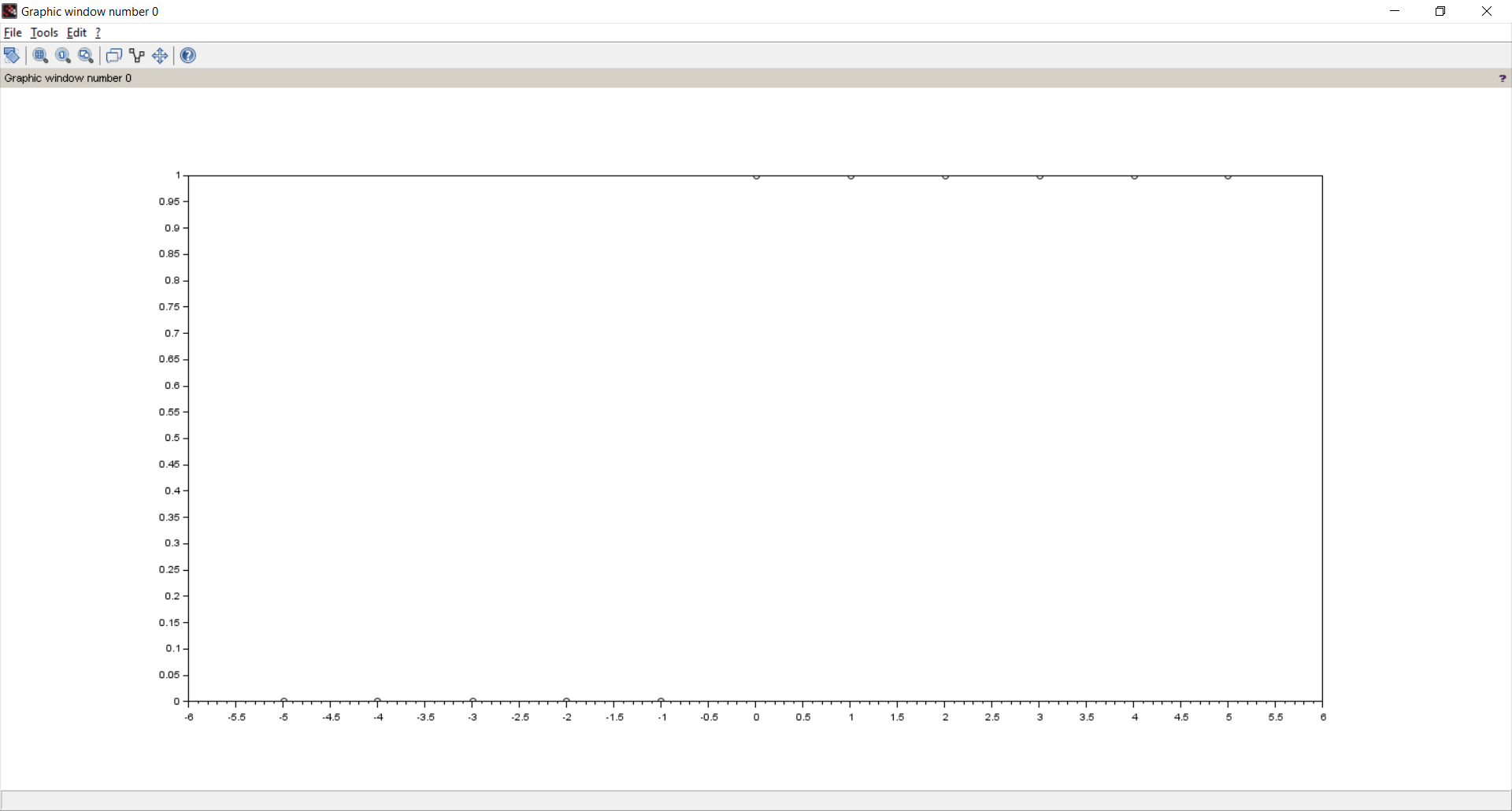
close;

x=[-1 -2 -3 -4 -5 0 1 2 3 4 5];

y=[0 0 0 0 0 1 1 1 1 1 1];

plot(x,y,'ko');

OUTPUT:-



**Experiment – 6b**

**Aim –** Write a program to plot a square wave function*.*

*//Program to generate a square wave*

clc;

clear;

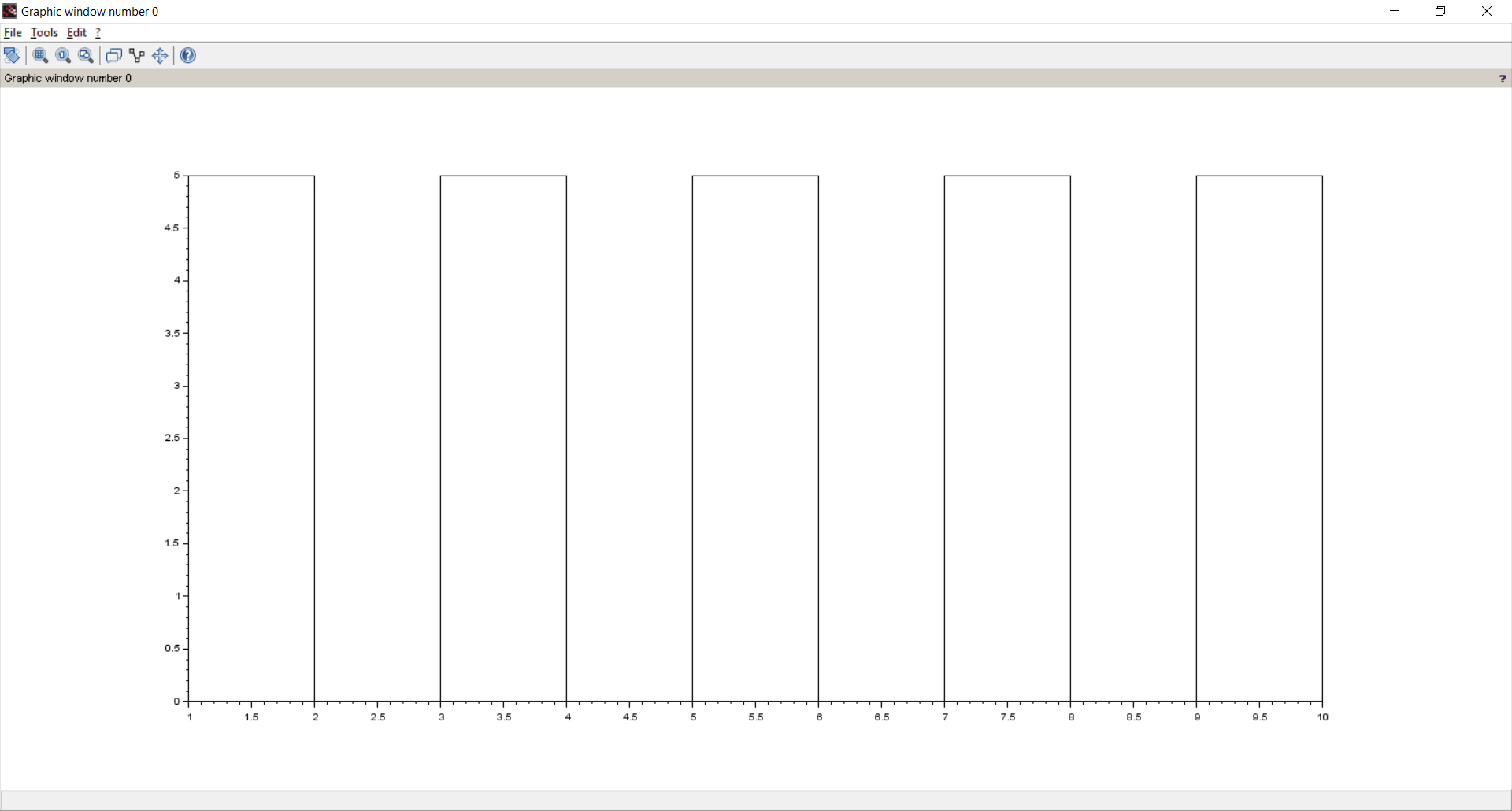
close;

x=[1 2 3 4 5 6 7 8 9 10];

y=[5 0 5 0 5 0 5 0 5 0];

plot2d2(x,y);

OUTPUT:-



**Experiment – 7a**

**Aim –** Write a program to find solution of a non-linear equation using Bisection method

*//Bisection Method*

clc;

clear;

close;

deff('y=f(x)','y=x^3-4\*x-9')

x1=2;

x2=3;

e=0.001;

i=0;

printf('f(x)=x^3-4\*x-9\n')

printf('Iteration\tx1\t\tx2\t\tz\t\tf(z)\n')

while abs(x1-x2)>2\*e

    z=(x1+x2)/2

printf(' %i\t\t%f\t%f\t%f\t%f\n',i,x1,x2,z,f(z))

if f(z)\*f(x1)>0

x1=z

else

x2=z

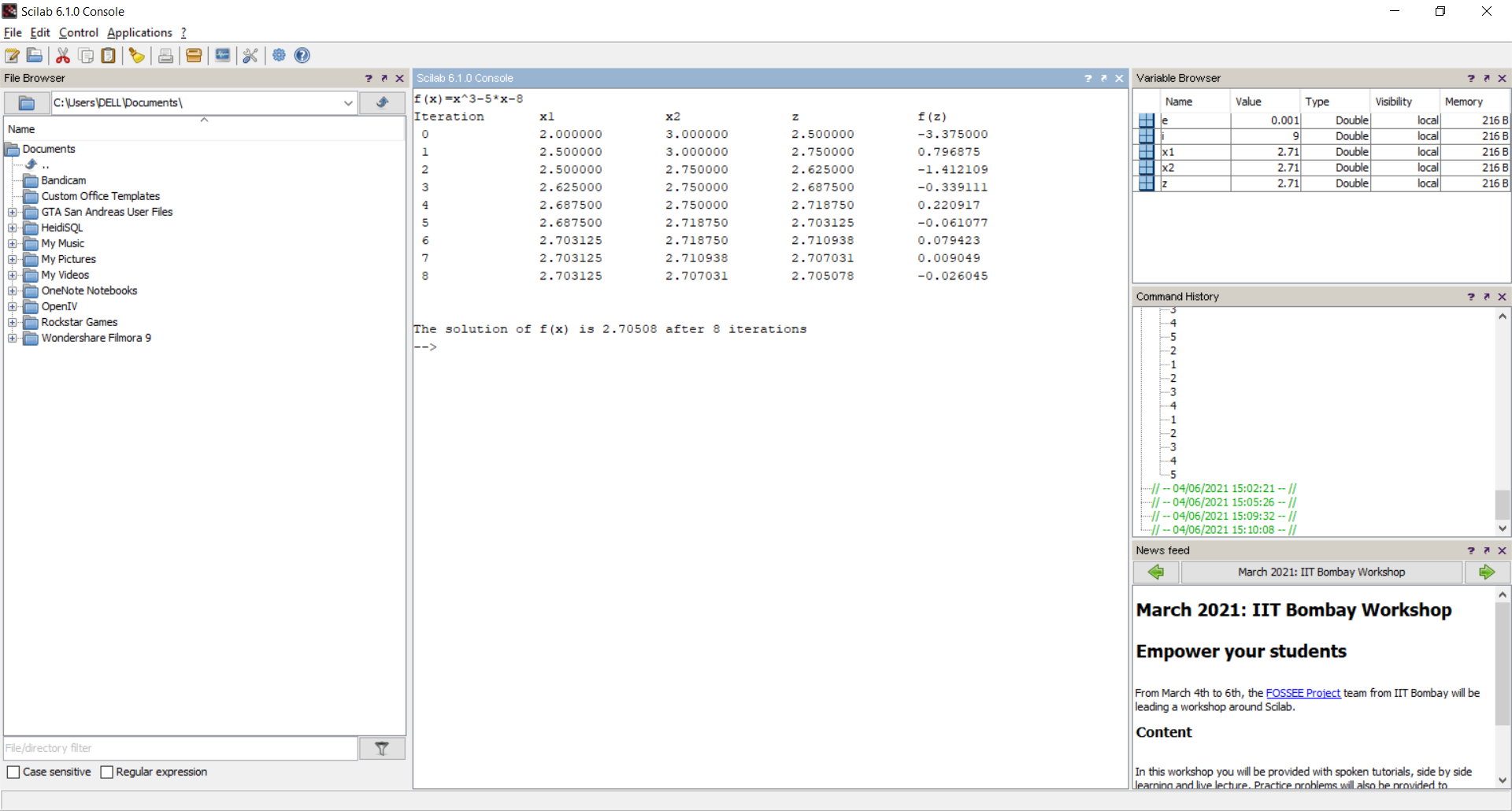
end

    i=i+1

end

printf('\n\nThe solution of f(x) is %g after %i iterations',z,i-1)

OUTPUT:-



**Experiment – 7b**

**Aim –** Write a program to find solution of an non-linear equation using Newton - Raphson method

*//Newton-Raphson Method*

clc;

clear;

close;

deff('x=f(x)','x=cos(x)-x\*exp(x)')

deff('x=f1(x)','x=-sin(x)-(x+1)\*exp(x)')

x0=0.5;

e=0.00001;

printf('\nf(x)=cos(x)-x\*exp(x)')

printf('\nf1(x)=-sin(x)-(x+1)\*exp(x)\n')

printf(' n\t xn\t\t f(xn)\t\t f1(xn)\t\t xn+1\t\t Error\n\n')

for i=1:4

x1=x0-f(x0)/f1(x0)

    e1=abs(x0-x1)

printf(' %i\t%f\t%f\t%f\t%f\t%f\n',i-1,x0,f(x0),f1(x0),x1,e1)

x0=x1;

if(e1<e)

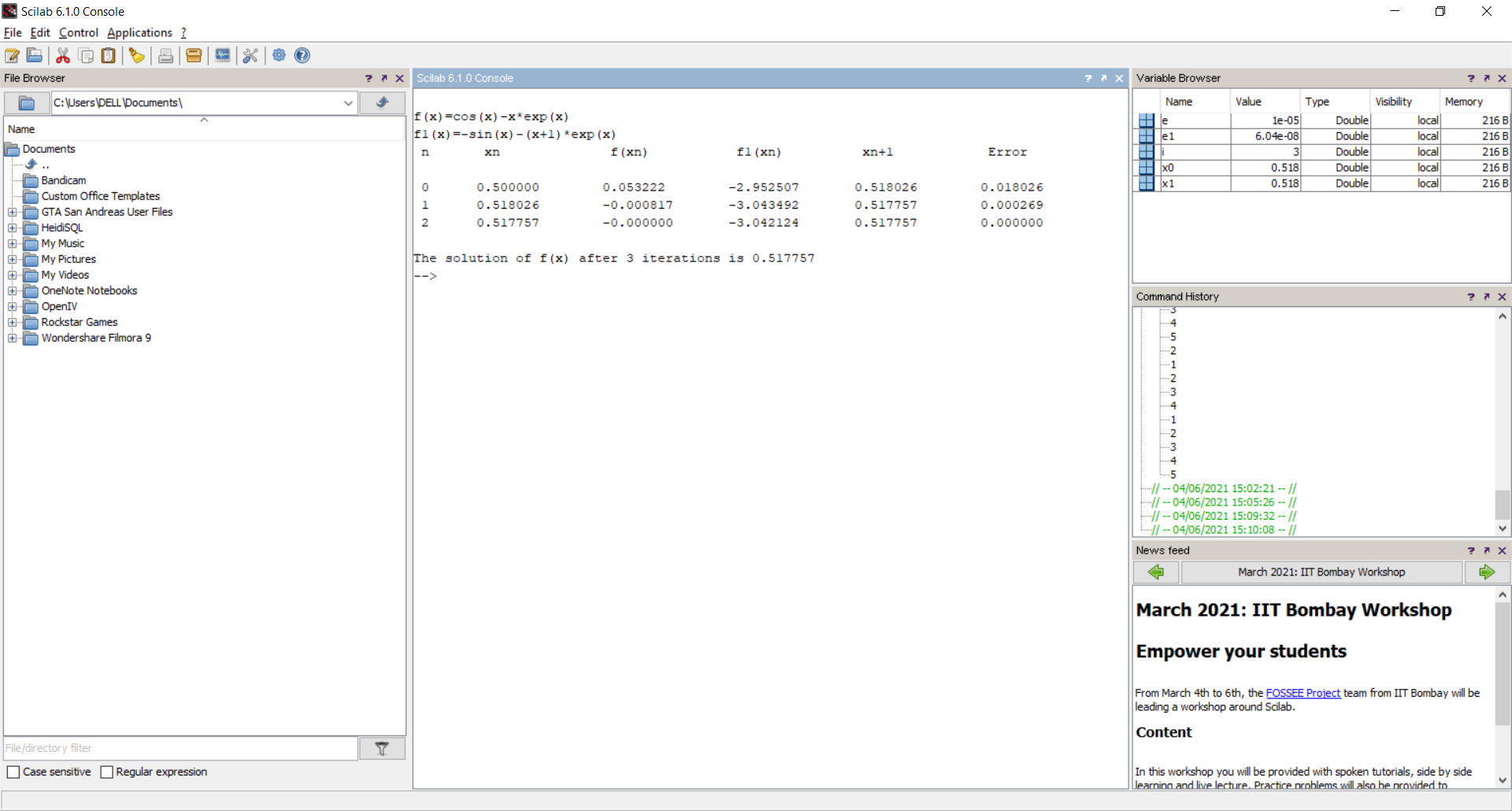
break;

end

end

printf( '\nThe solution of f(x) after %i iterations is %f',i,x1)

OUTPUT:-



**Experiment – 8a**

**Aim –** Write a program to evaluate a definite integral using Trapezoidal rule

*//Program to find value of integral using Trapezoidal rule*

clc;

clear;

close;

deff('y=f(x)','y=1/(1+x^2)')

x0=0;

xn=6;

n=6;

h=(xn-x0)/n;

s=0;

for i=1:n

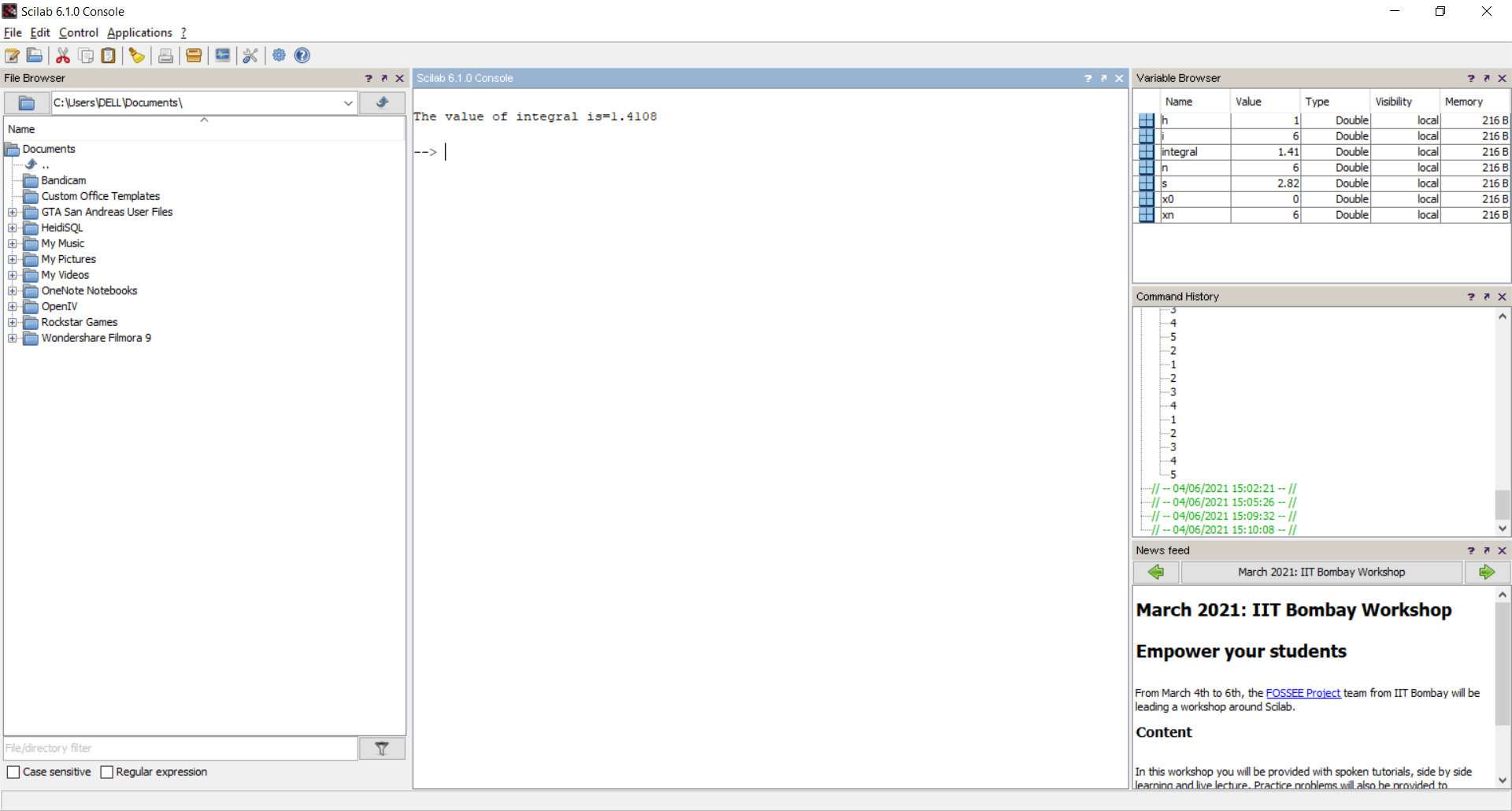
    s=s+f(x0+(i-1)\*h)+f(x0+i\*h);

end

integral=(h\*s)/2;

printf('\nThe value of integral is=%g\n',integral)

OUTPUT:-



**Experiment – 8b**

**Aim –** Write a program to evaluate a definite integral using Simpson’s one third rule

*//Program to find value of integral using Simpson's one third rule*

clc;

clear;

close;

deff('y=f(x)','y=sin(x)')

x0=0;

xn=%pi;

n=10;         *//n should be even*

h=(xn-x0)/n;

s=0;

for i=1:2:n

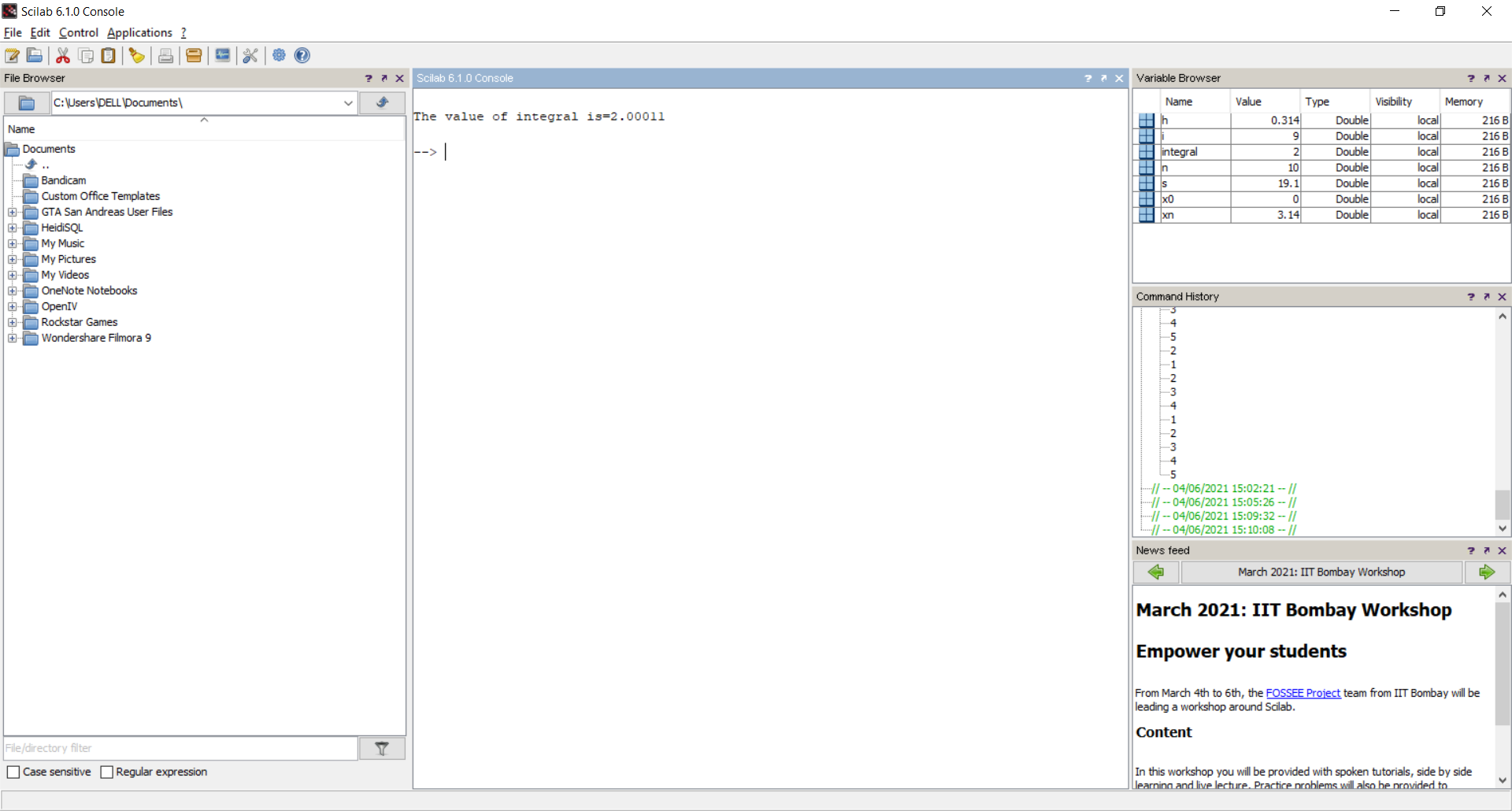
    s=s+f(x0+(i-1)\*h)+4\*f(x0+i\*h)+f(x0+(i+1)\*h);

end

integral=(h\*s)/3;

printf('\nThe value of integral is=%g\n',integral)

OUTPUT:-



**Experiment – 8c**

**Aim –** Write a program to evaluate a definite integral using Simpson’s three eighth rule

*//Program to find value of integral using Simpson's three eighth rule*

clc;

clear;

close;

deff('y=f(x)','y=1/(1+x^2)')

x0=0;

xn=6;

n=6;     *//n should be a multiple of three*

h=(xn-x0)/n;

s=0;

for i=1:3:n

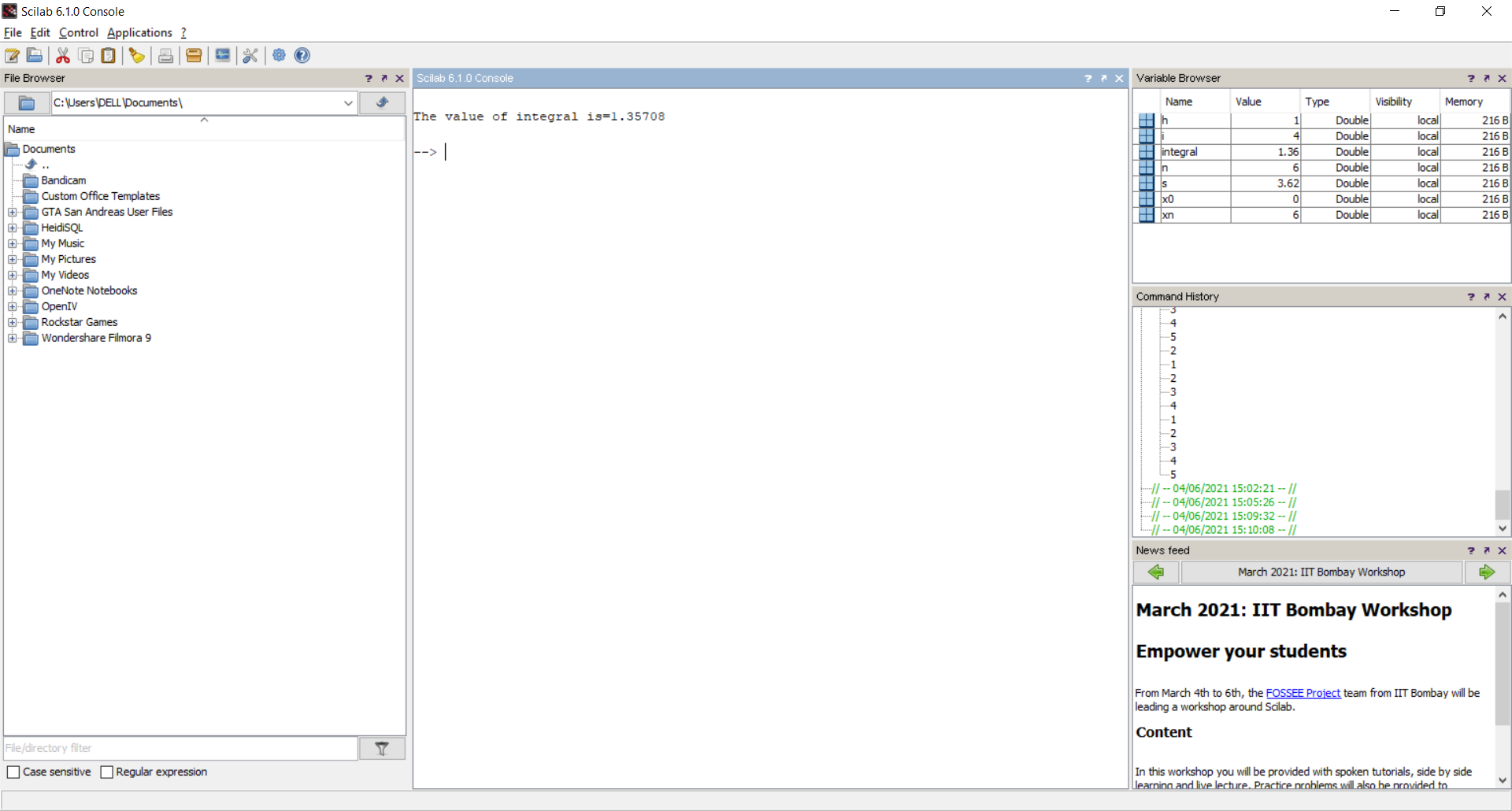
    s=s+f(x0+(i-1)\*h)+3\*f(x0+i\*h)+3\*f(x0+(i+1)\*h)+f(x0+(i+2)\*h);

end

integral=(3\*h\*s)/8;

printf('\nThe value of integral is=%g\n',integral)

OUTPUT:-



**Experiment – 9**

**Aim –** Write a program to find the initial value problem using Euler’s method

*//Program to solve the equation y'=f(x,y); y(x0)=0 for y(xn) using Euler's*

*//method*

clc;

clear;

close;

deff('z=f(x,y)','z=(y-x)/(y+x)')

x0=0;

y0=1;

xn=0.1;

h=0.02;

x=x0;

y=y0;

while x~=xn

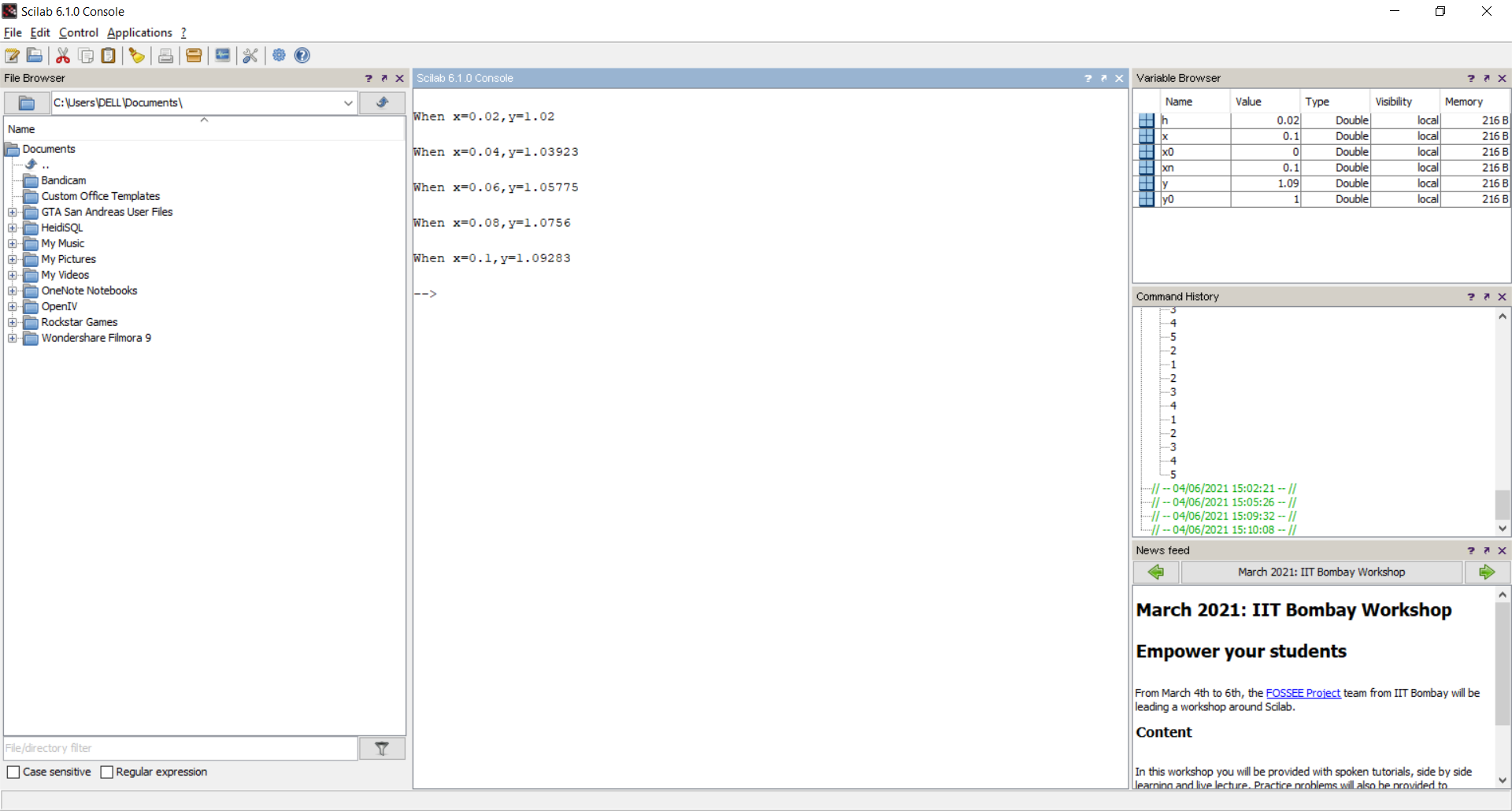
    y=y+h\*f(x,y);

    x=x+h;

printf('\nWhen x=%g,y=%g\n',x,y);

end

OUTPUT:-



**Experiment – 10**

**Aim –** Write a program to find the solution of initial value using Runge-Kutta method of fourth order

*//Program to solve the equation y'=f(x,y);y(x0)=y0 for y(xn) using Runge-Kutta*

*//method of fourth order*

clc;

clear;

close;

deff('z=f(x,y)','z=x\*x-y')

x0=0;

y0=1;

xn=0.2;

h=0.1;

x=x0;

y=y0;

while x~=xn

    k1=h\*f(x,y);

    k2=h\*f(x+h/2,y+k1/2);

    k3=h\*f(x+h/2,y+k2/2);

    k4=h\*f(x+h,y+k3);

    k=(k1+(k2+k3)\*2+k4)/6;

    x=x+h;

    y=y+k;

printf('\nWhen x=%g,y=%g\n',x,y)

end

OUTPUT:-

