***APPLIED MATHEMATICS***

***LAB***

***ETMA-252***

**Faculty Name: - Mr. PANKAJ SIKARWAR Student Name: - ARIHANT JAIN**

**Roll No.: - 08914802719**

**Semester: - 4th SEMESTER**



Maharaja Agrasen Institute of Technology, PSP Area, Sector – 22, Rohini,

New Delhi – 110085

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **EXPERIMENTS** | **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 | 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.*

*• Multiplication of two matrices.*

*• Transpose of a matrix*

**#Code**

***(Addition of two matrices.)***

clc;

clear ;

x= input("Name : ARIHANT JAIN")

y= input("Roll Num : 08914802719")

*r=input('Enter The Number Of Rows:')*

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

A=zeros(r,c); B=zeros(r,c); C=zeros(r,c)

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

for i=1:r for j=1:c

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

end end

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

for i=1:r for j=1:c

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

end end

for i=1:r for j=1:c

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

end end

disp('First Matrix==>')

disp(A)

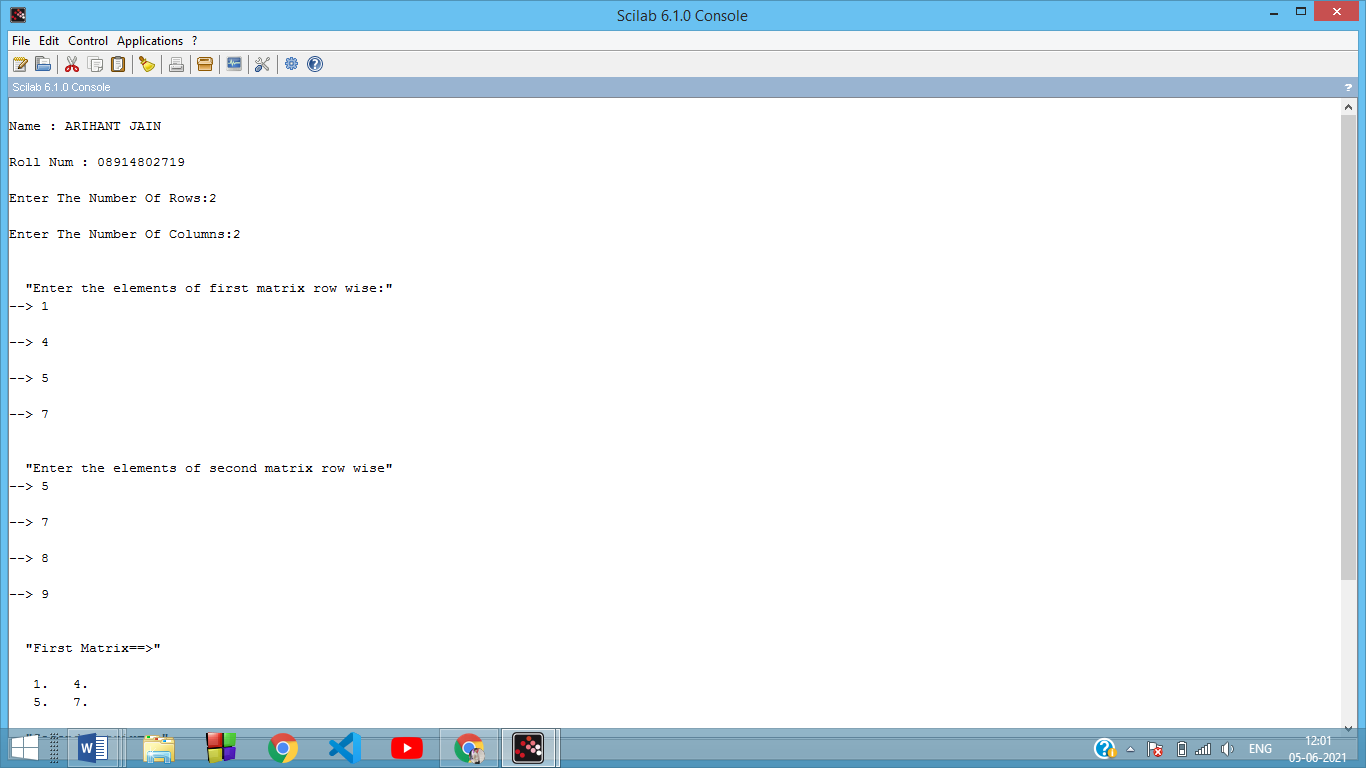
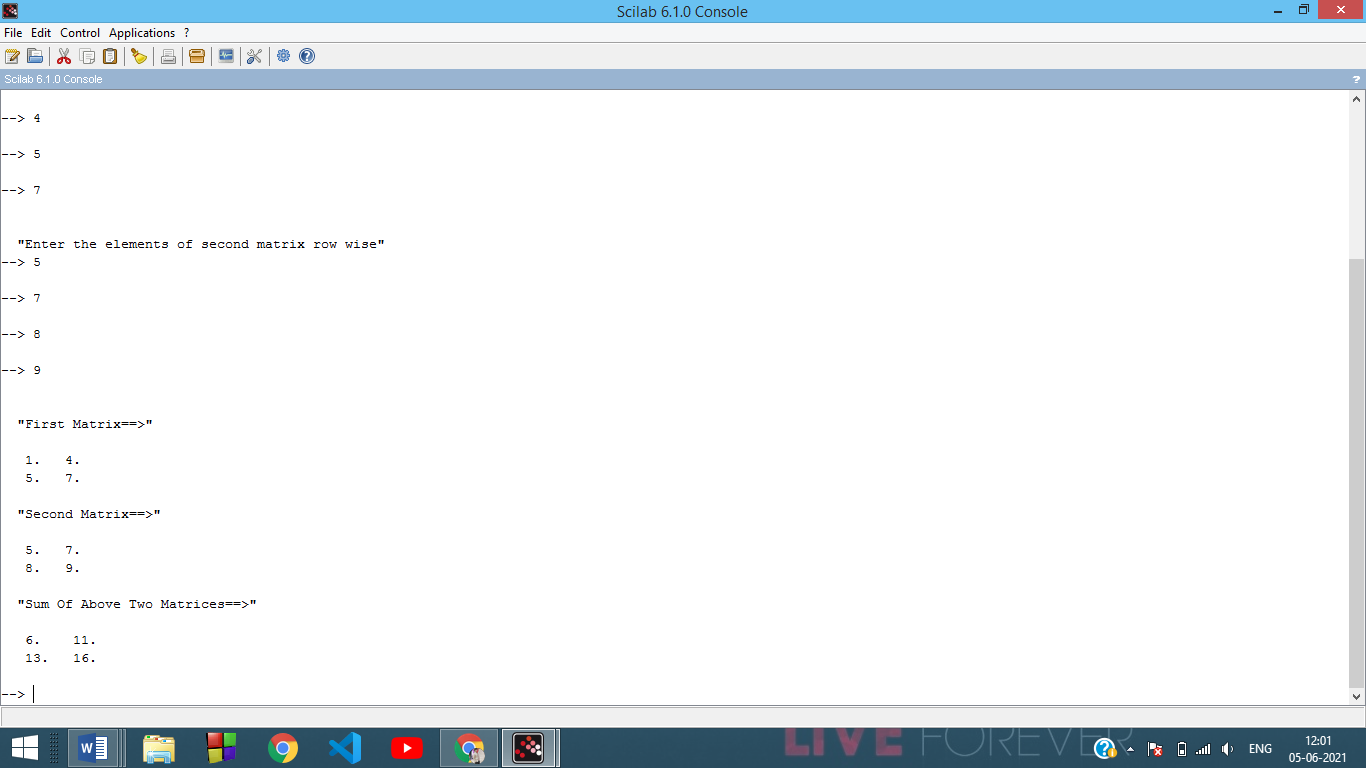
disp('Second Matrix==>')

disp(B)

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

disp(C)

***OUTPUT:***

**#Code**

**(*Multiplication of two matrices)***

*//matrix multiplication*

clc;

clear ;

x= input("Name : ARIHANT JAIN")

y= input("Roll Num : 08914802719")

*//matrix multiplication*

m=input("Enter number of rows for A:")

n=input("Enter number of columns for A:")

a=zeros(m,n)

disp("Enter the elements of matrix A:");

for i=1:m

for j=1:n

a(i,j)=input("");

end;

end;

o=input("Enter number of rows for B:")

p=input("Enter number of columns for B:")

b=zeros(o,p)

disp("Enter the elements of matrix B:");

for i=1:o

for j=1:p

b(i,j)=input("");

end;

end;

disp("Matrix A:");

disp(a);

disp("Matrix B:");

disp(b);

if n==o then

c=zeros(m,p)

for i=1:m

for j=1:p

for k=1:n

c(i,j)=c(i,j)+a(i,k)\*b(k,j);

end;

end;

end;

disp("Multiplication Matrix C:");

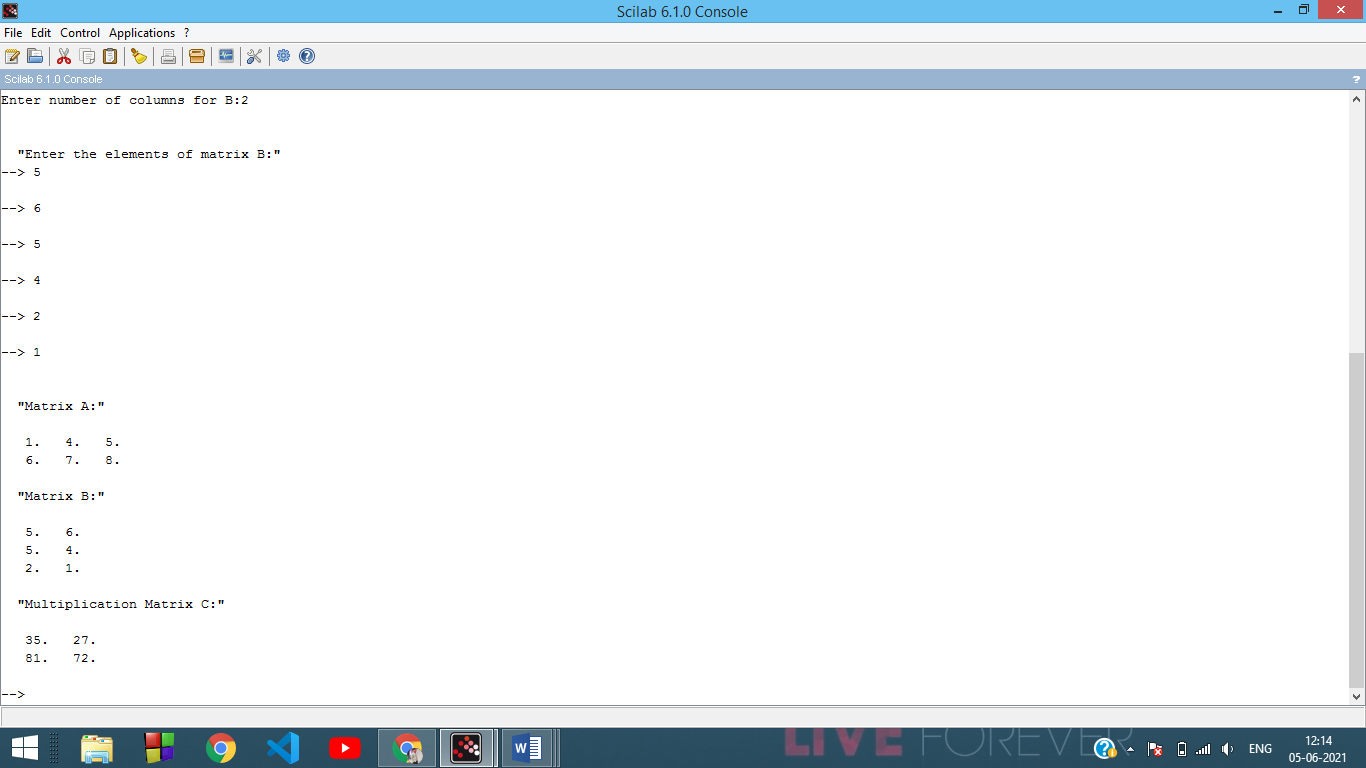
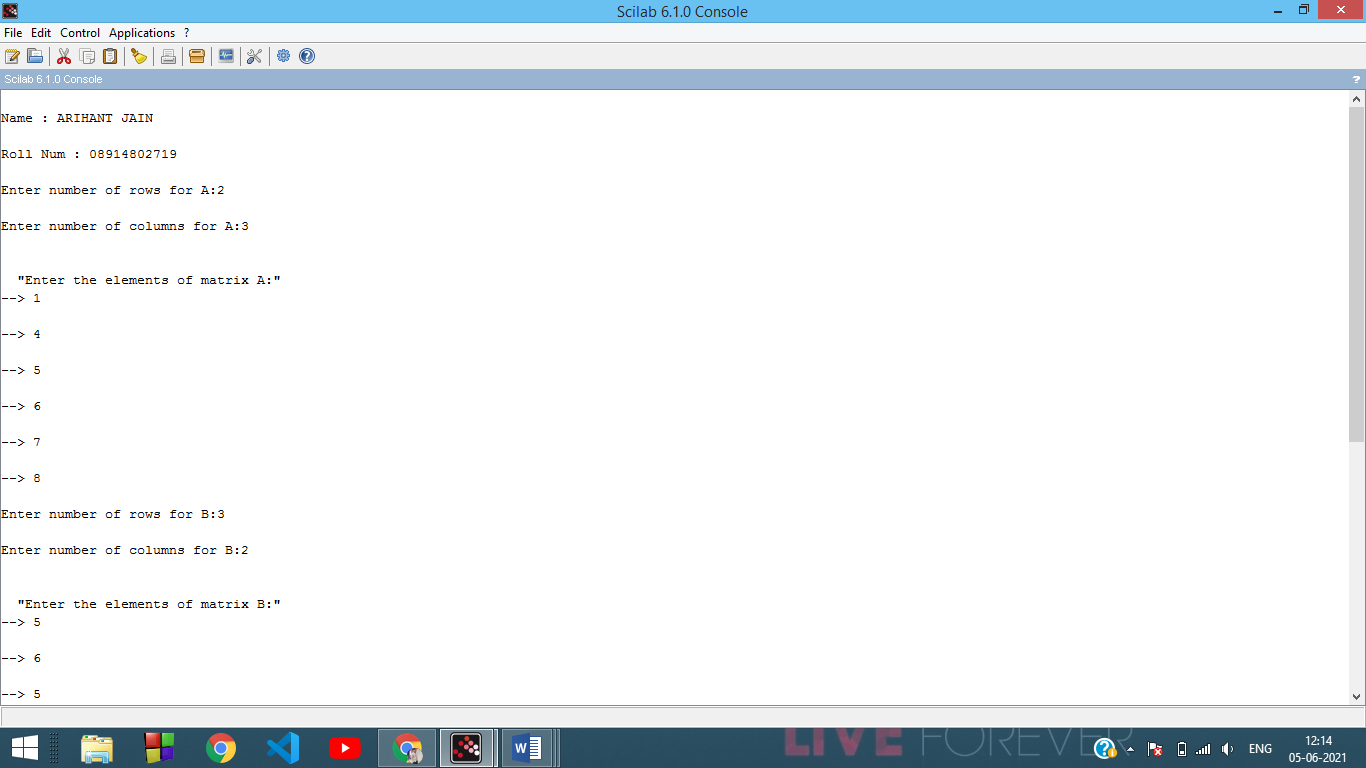
disp(c);

else

disp("Matrix cannot be multiplied");

end;

***OUTPUT:***



**(*Transpose of a matrix*)**

**Code**

clc;

clear ;

x= input("Name : ARIHANT JAIN")

y= input("Roll Num : 08914802719")

m=input("Enter number of rows:")

n=input("Enter number of columns:")

a=zeros(m,n)

b=zeros(n,m)

disp("Enter the elements of matrix A:");

for i=1:m

for j=1:n

a(i,j)=input("");

end;

end;

for i=1:m

for j=1:n

b(j,i)=a(i,j);

end;

end;

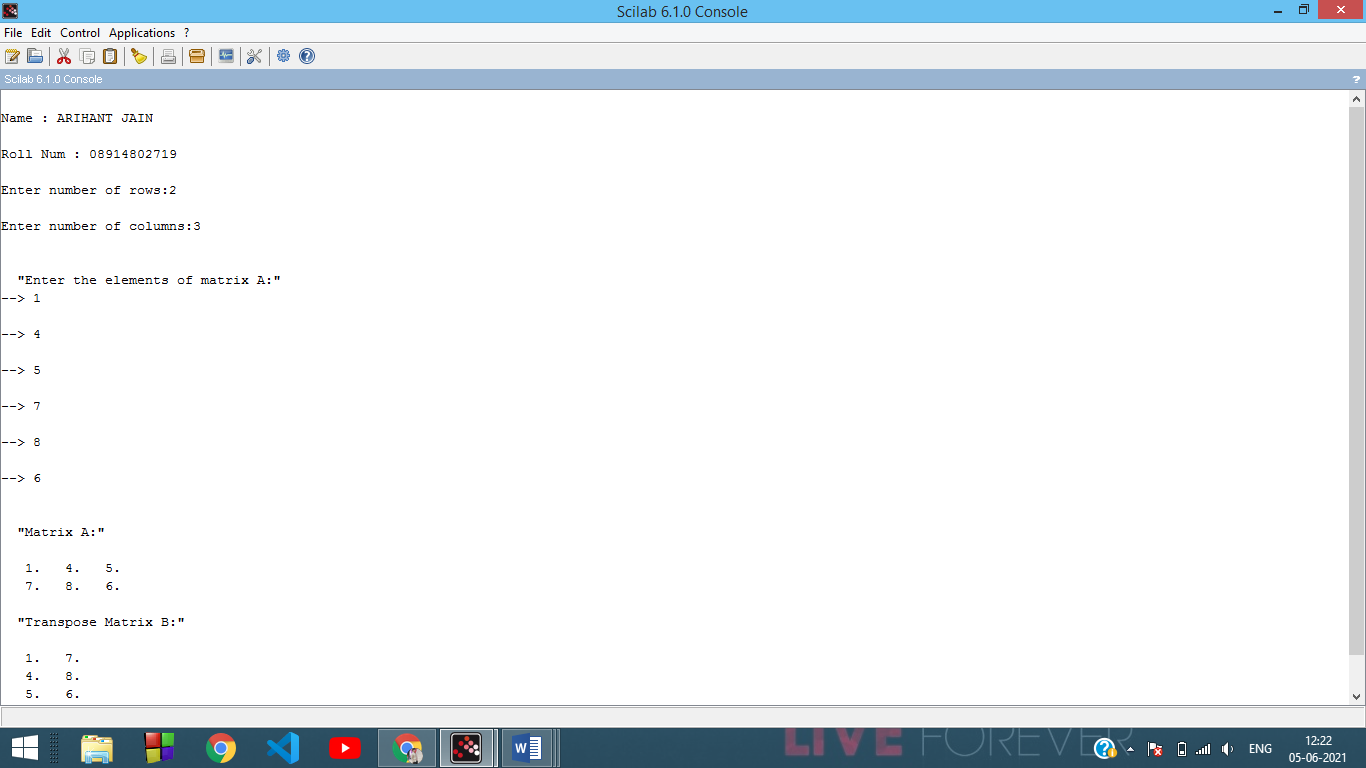
disp("Matrix A:");

disp(a);

disp("Transpose Matrix B:");

disp(b);

***OUTPUT:***



**Experiment – 2**

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

**#Code**

clc;

clear ;

x= input("Name : ARIHANT JAIN")

y= input("Roll Num : 08914802719")

*// Inverse of a 3 by 3 matrix using gauss jordan Method*

disp('Enter a 3 by 3 matrix row-wise, make sure that diagonal elements are non -zeros')

for i=1:3

for j=1:3

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

end

end

disp('Entered Matrix is')

disp(A)

if det(A)==0

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

break;

end

*//Taking the augmented matrix [A|I],*

B=[A eye(3,3)]

disp('Augumented matrix is:')

disp(B)

*//Making B(1,1)=1*

B(1,:) = B(1,:)/B(1,1);

*//Making B(2,1) and B(3,1)=0*

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

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

*//Making B(2,2)=1 and B(1,2), B(3,2)=0*

B(2,:) = B(2,:)/B(2,2);

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

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

*// Making B(3,3)=1 and B(1,3), B(2,3)=0*

B(3,:) = B(3,:)/B(3,3);

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

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

disp('Augumented matrix after row operations is:')

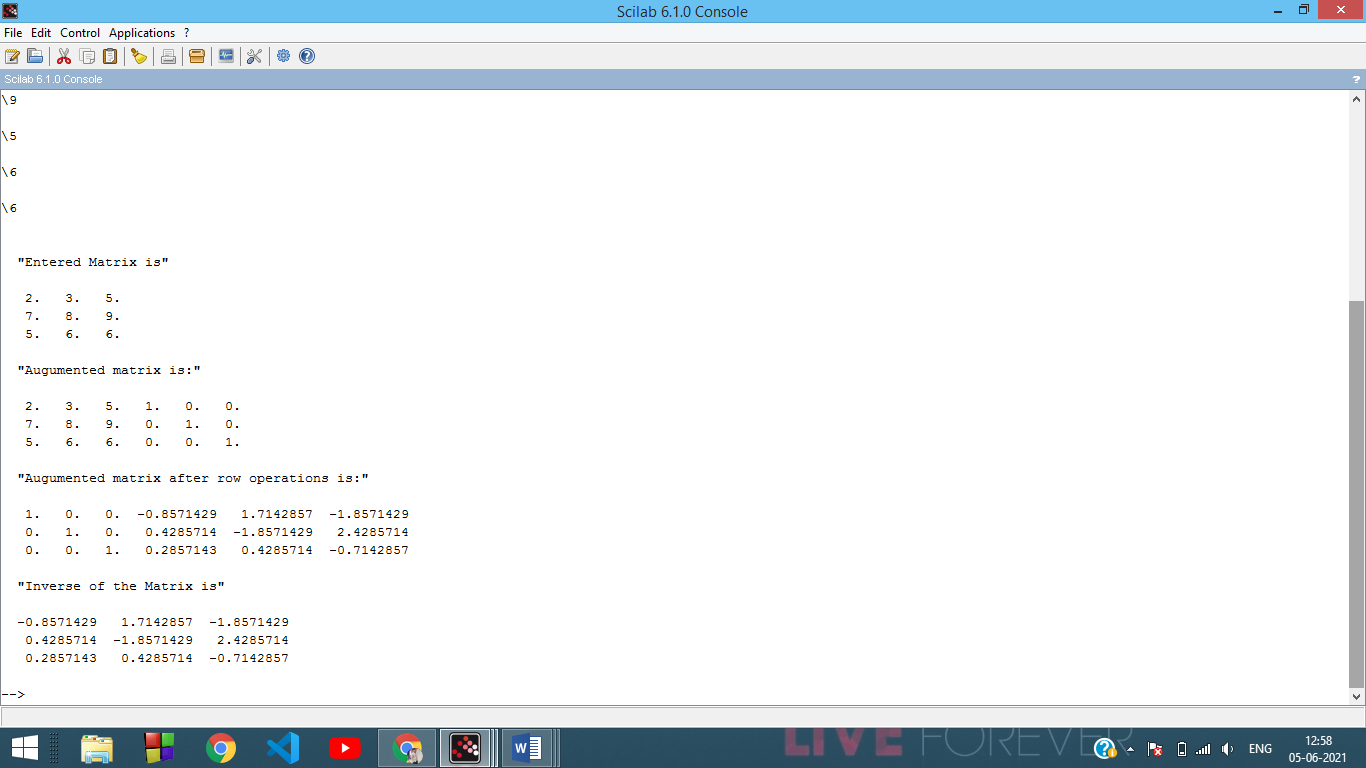
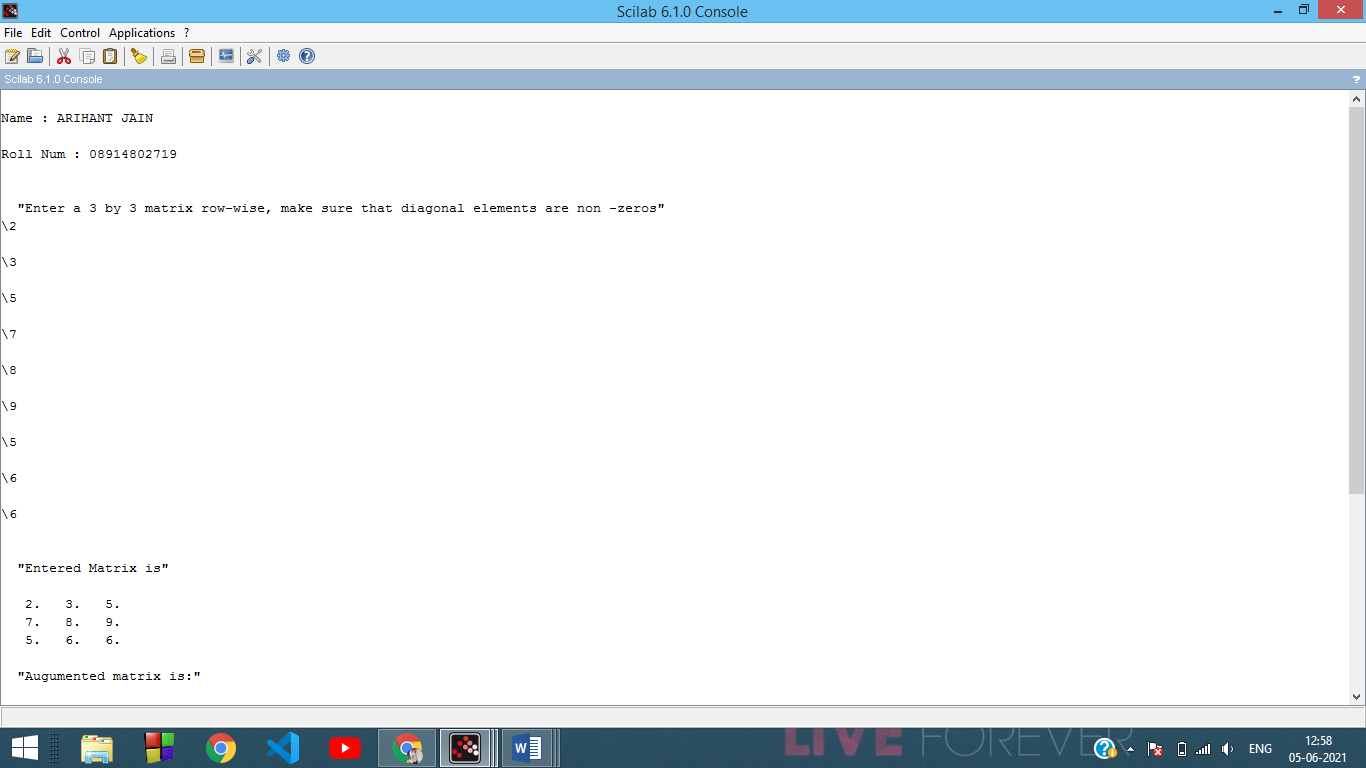
disp(B)

B(:,1:3)=[]

disp('Inverse of the Matrix is')

disp(B)

***OUTPUT:***



**Experiment – 3**

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

**#Code**

clc;

clear ;

x= input("Name : ARIHANT JAIN")

y= input("Roll Num : 08914802719")

*//Eigen values and Eigen vectors*

disp('Enter the 2x2 matrix:')

for i=1:2 for j=1:2

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

end end

disp('The Matrix entered is :')

disp(A)

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);

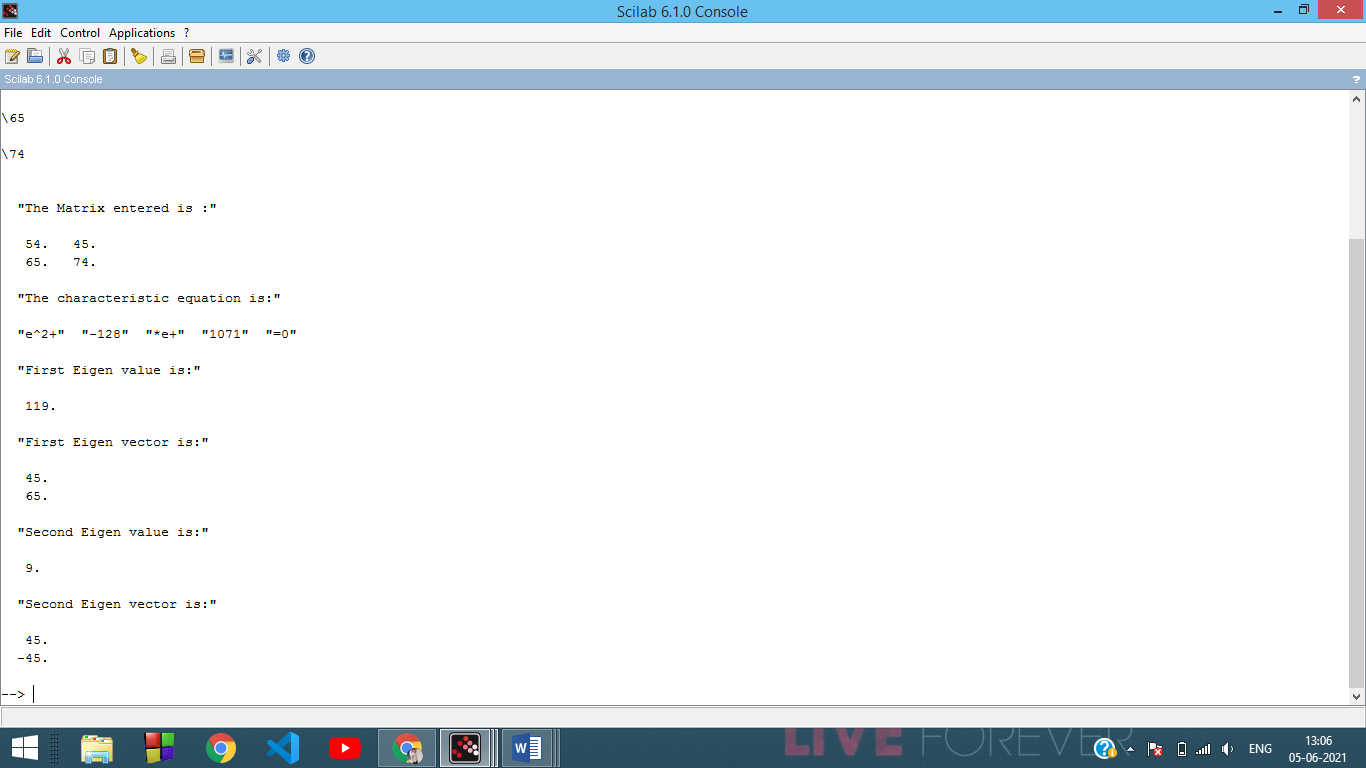
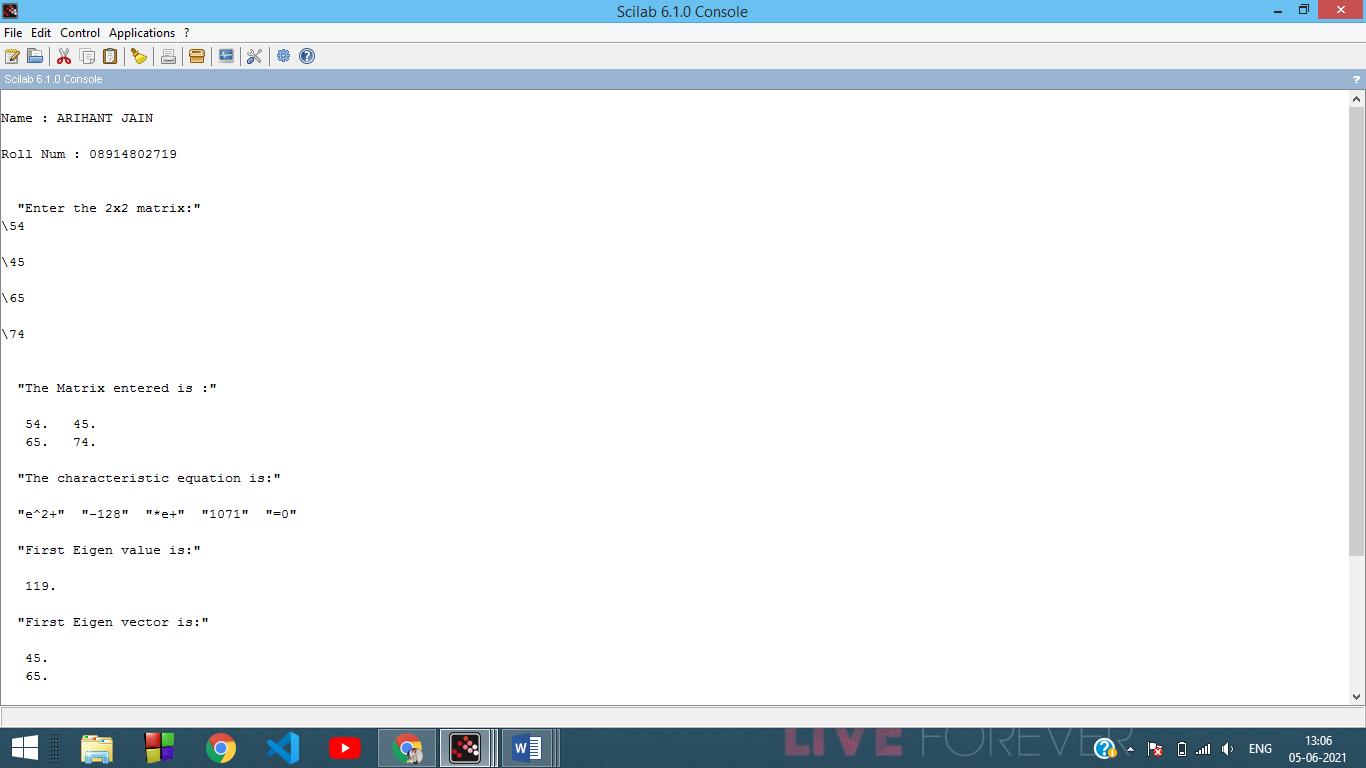
disp('Second Eigen value is:');

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.

**#Code**

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

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*

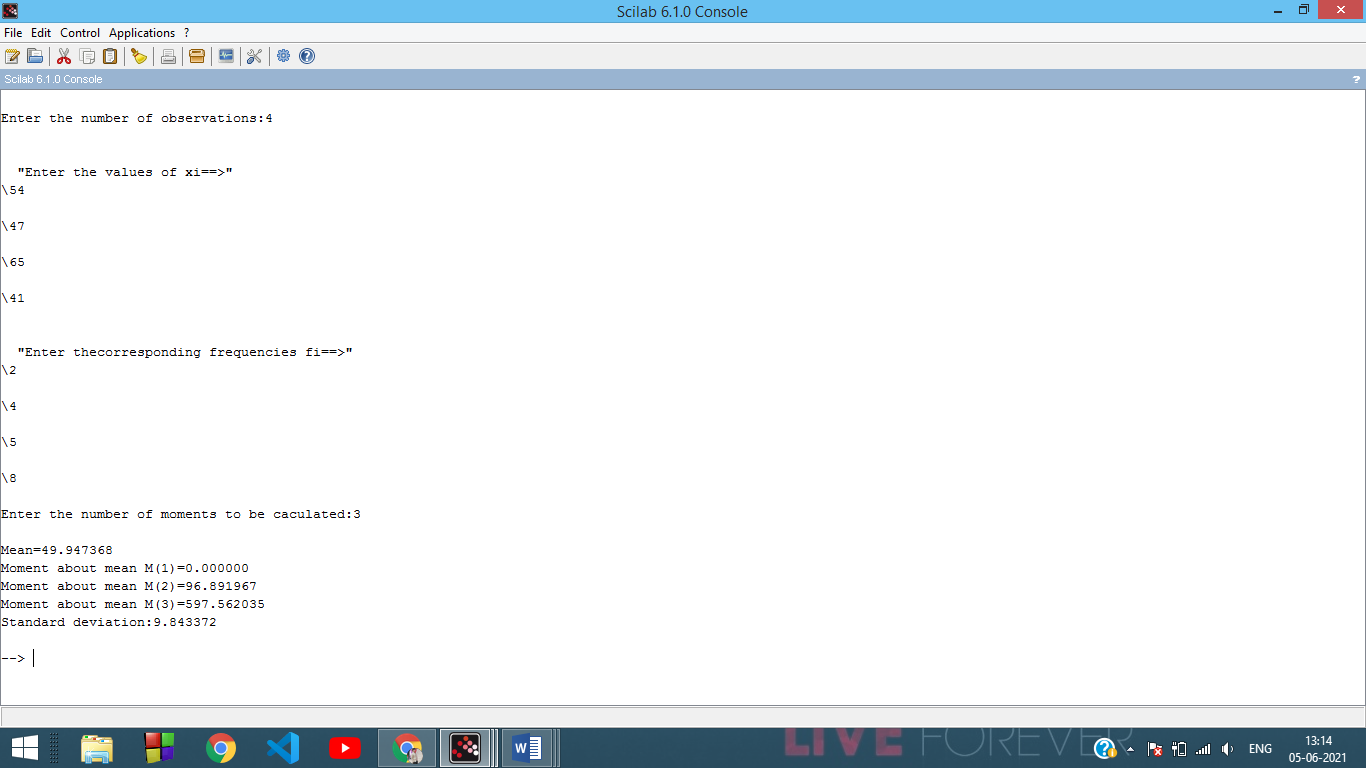
printf('Moment about mean M(%d)=%f\n',j,M(j));

end;

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

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

***OUTPUT:***



**Experiment - 5**

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

**#Code**

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

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

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

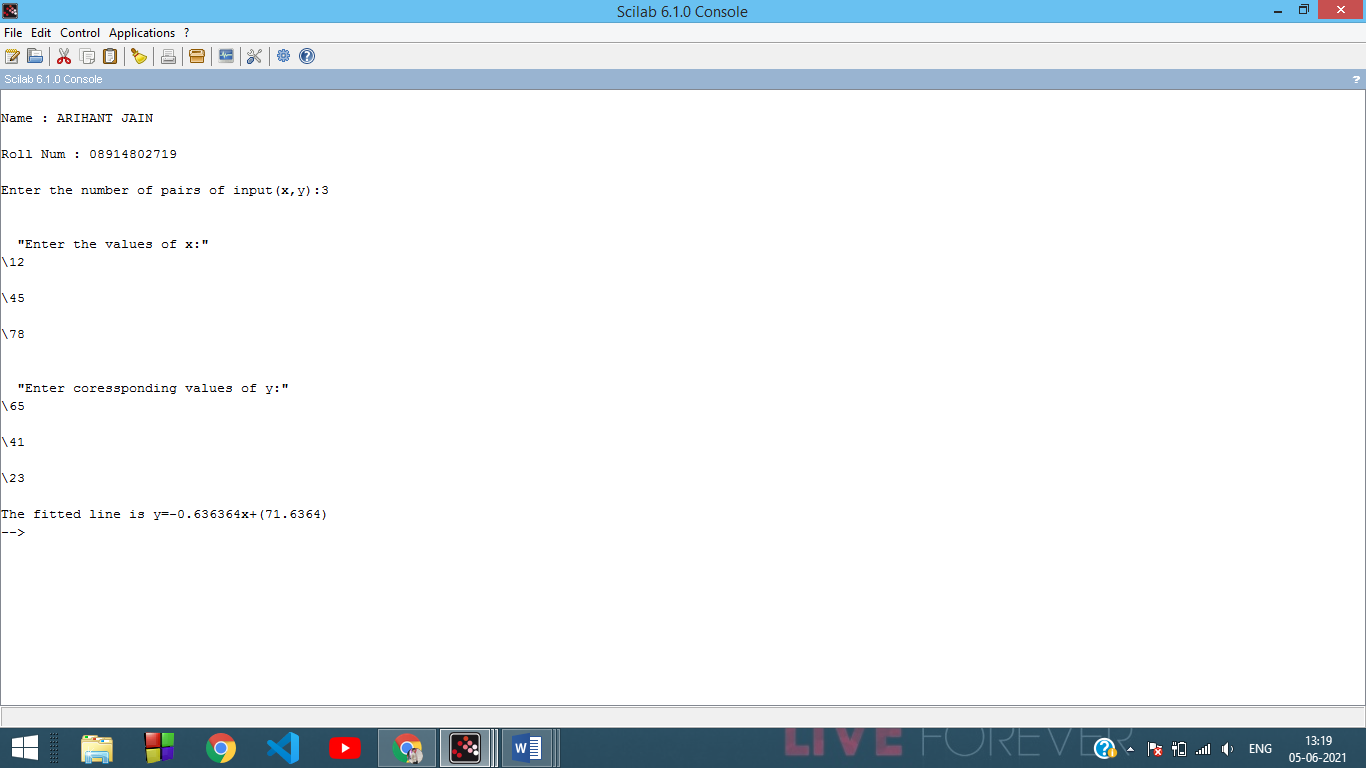
s4=s4+x(i)\*y(i);

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 - 6**

**Aim -** ***Write a program to plot***

*• unit step function*

*• square wave function*

**#Code**

***(unit step function)***

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

t = -6:0.01:6;

u = ones ( t ).\*( t >=0) ;

plot (t ,u ) ;

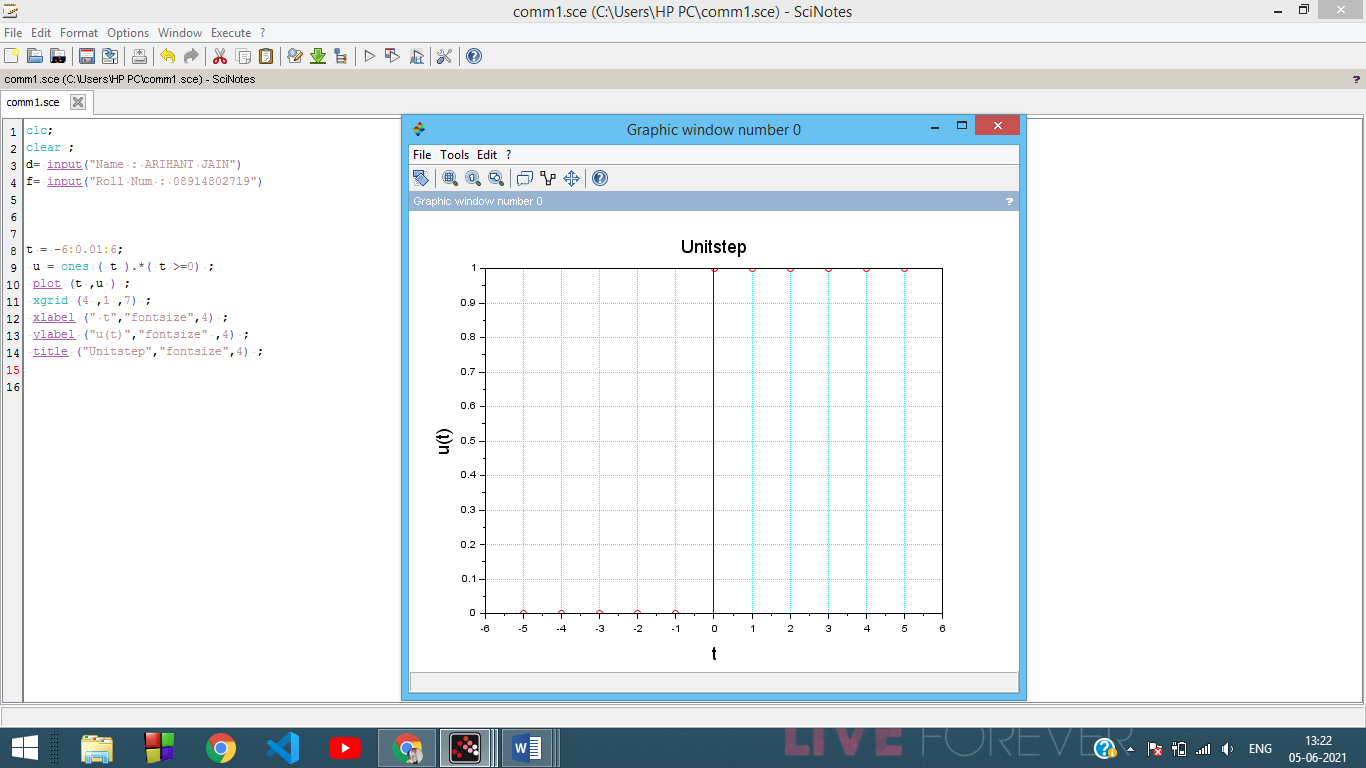
xgrid (4 ,1 ,7) ;

xlabel (" t","fontsize",4) ;

ylabel ("u(t)","fontsize" ,4) ;

title ("Unitstep","fontsize",4) ;

***OUTPUT:***



**#Code**

***(square wave function)***

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

*//Program to generate a square wave*

x=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];

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

plot2d2(x,y);

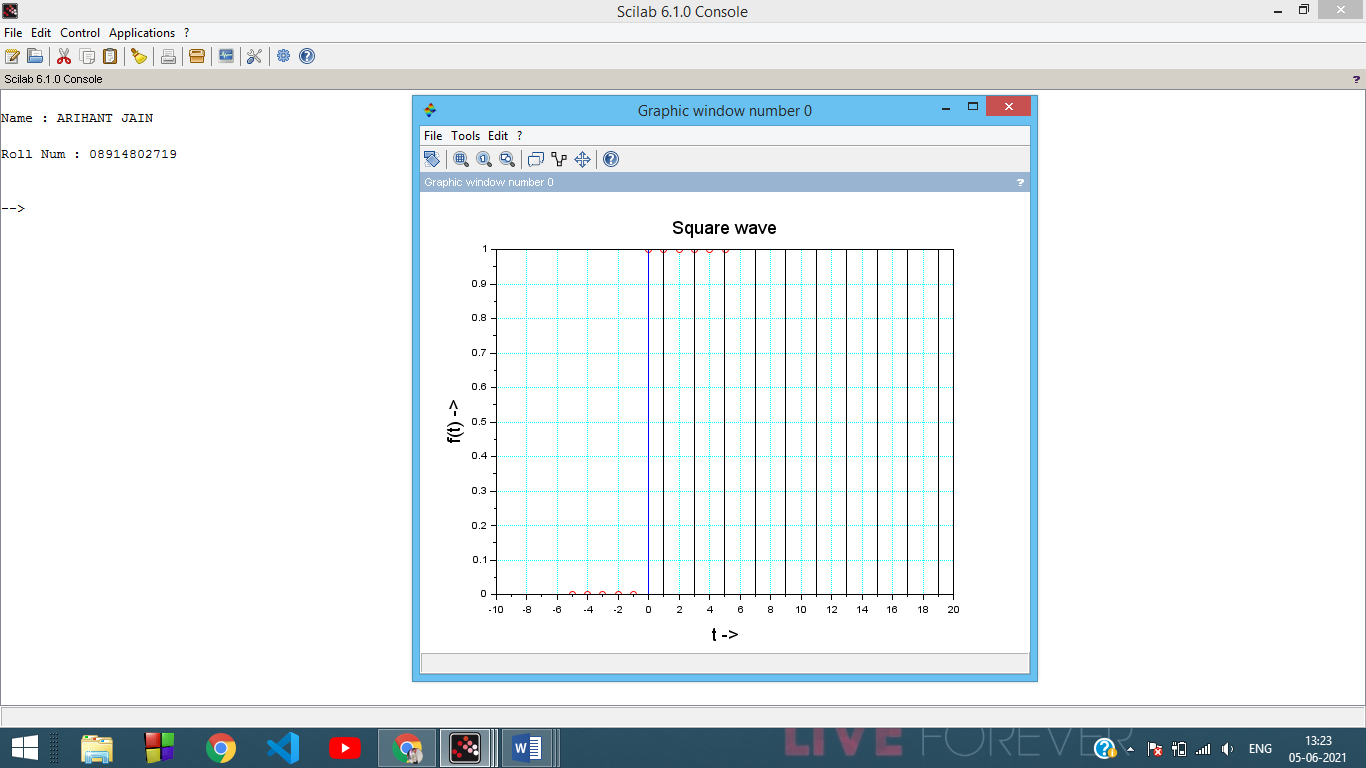
xgrid (4 ,1 ,7) ;

xlabel (" t ->","fontsize",4) ;

ylabel ("f(t) ->","fontsize" ,4) ;

title ("Square wave","fontsize",4) ;

***OUTPUT:***



**Experiment - 7**

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

* + ***Bisection method***
  + ***Newton - Raphson method***

**#Code**

**(*Bisection method*)**

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

*//Bisection Method*

deff('y=f(x)','y=x^3-9\*x+19')

x1=-4; x2=-3;

e=0.001;

i=0;

printf('f(x)=x^3-9\*x+19\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

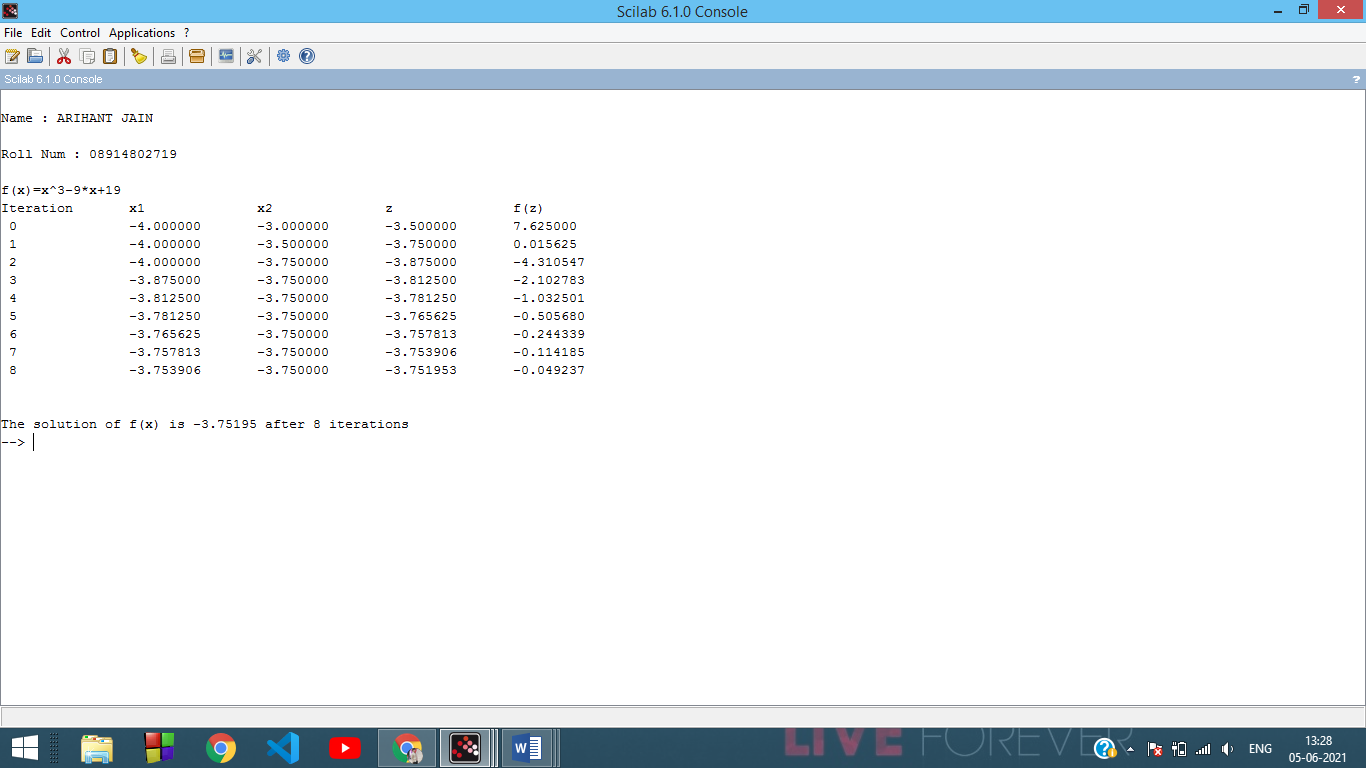
end

i=i+1

end

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

***OUTPUT:***



**#Code**

***(using Newton - Raphson)***

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

*//Newton-Raphson Method*

deff('y=f(x)','y=x^3+x^2-3\*x-3')

deff('y=df(x)','y=3\*x^2+2\*x-3')

x(1)=input('Enter Initial Guess:');

e= input("Answer correct upto : ");

for i = 1 : 100

x(i+1)=x(i)-f(x(i))/df(x(i));

err(i)=abs((x(i+1)-x(i))/x(i));

if err(i) < e

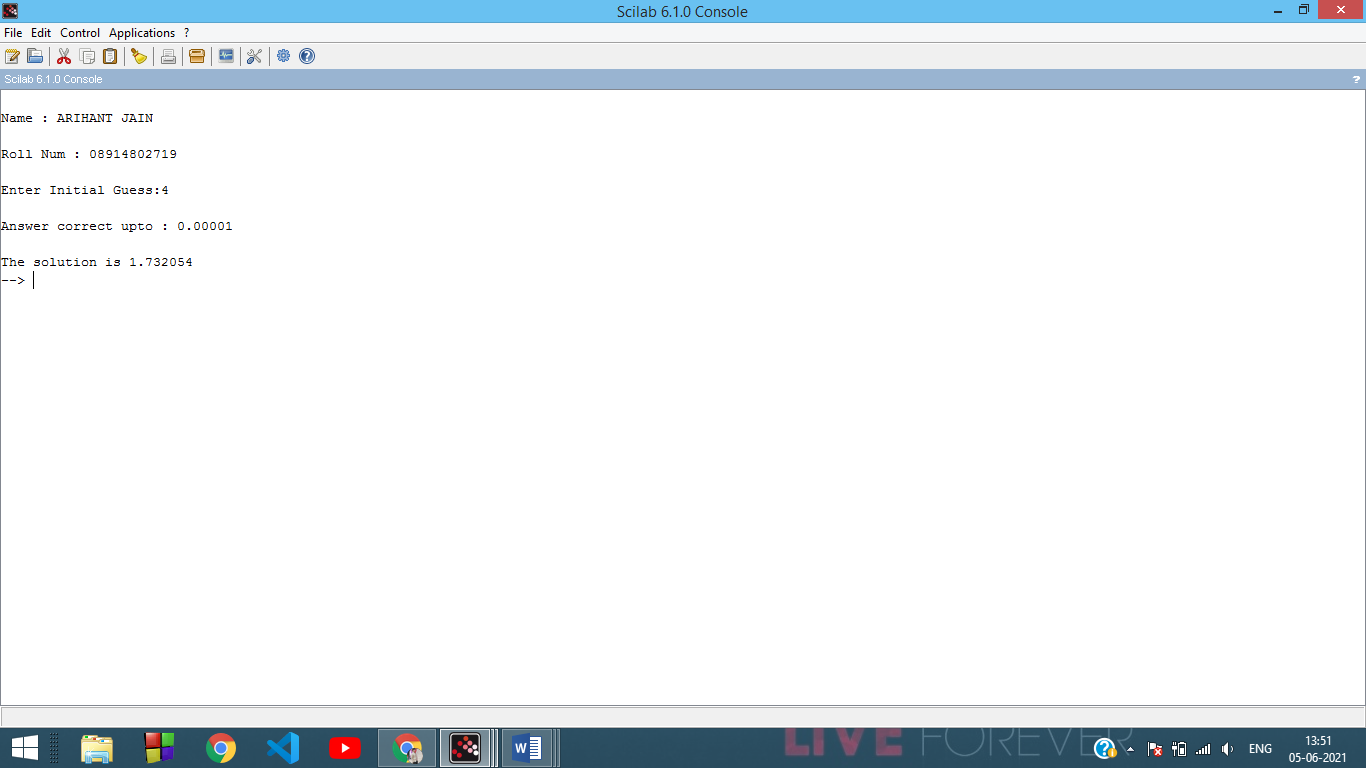
break;

end

end

printf('The solution is %f',x(i))

***OUTPUT:***



**Experiment - 8**

**Aim - Write a program to evaluate a definite integral using**

* + ***Trapezoidal rule***
  + ***Simpson’s one third rule***
  + ***Simpson’s three eighth rule***

**#Code**

***(Trapezoidal rule)***

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

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

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

printf('Integration of y=2/(1+x^2) from 0 to 6')

x0=0; xn=6;

n=6;

h=(xn-x0)/n;

s=0;

for i=1:n

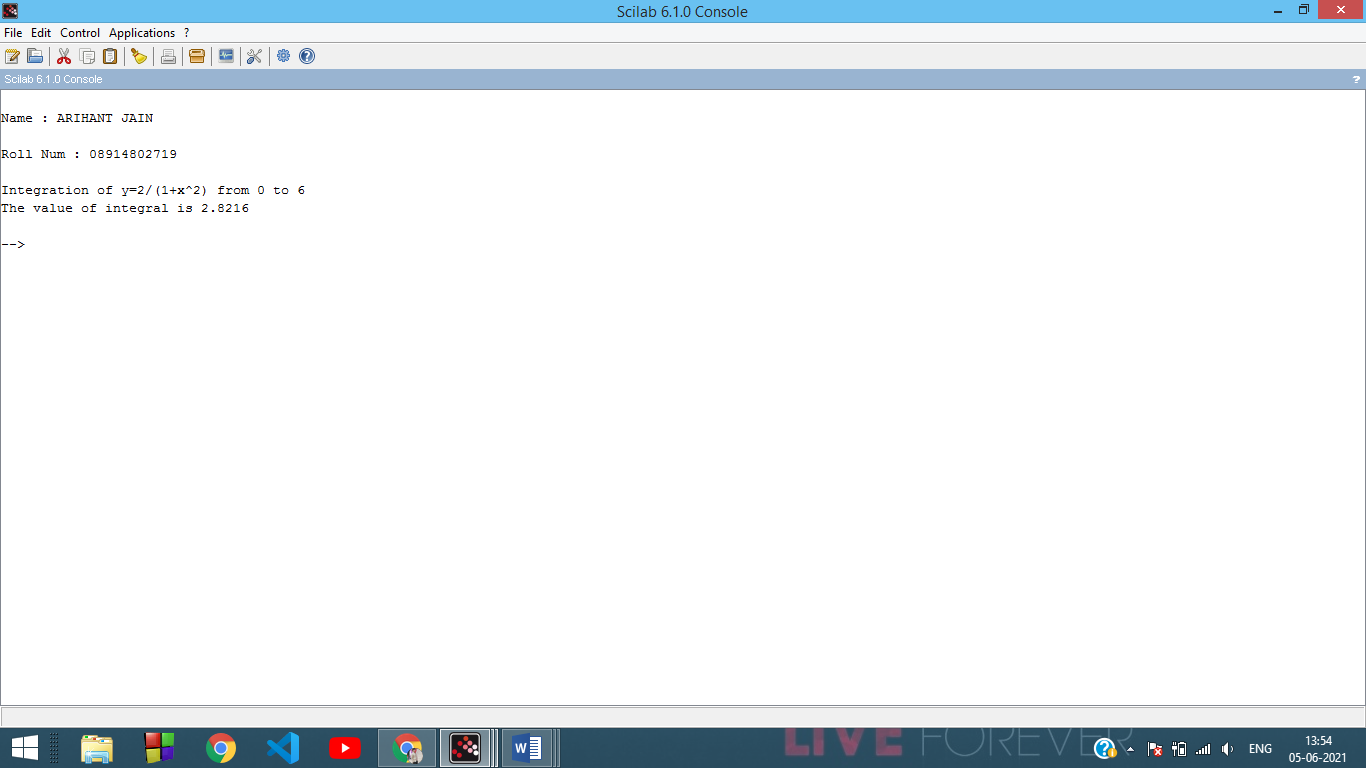
s=s+f(x0+(i-1)\*h)+f(x0+i\*h);

end

integral=(h\*s)/2;

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

***OUTPUT:***



**#Code**

***(using Simpson's one third rule)***

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

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

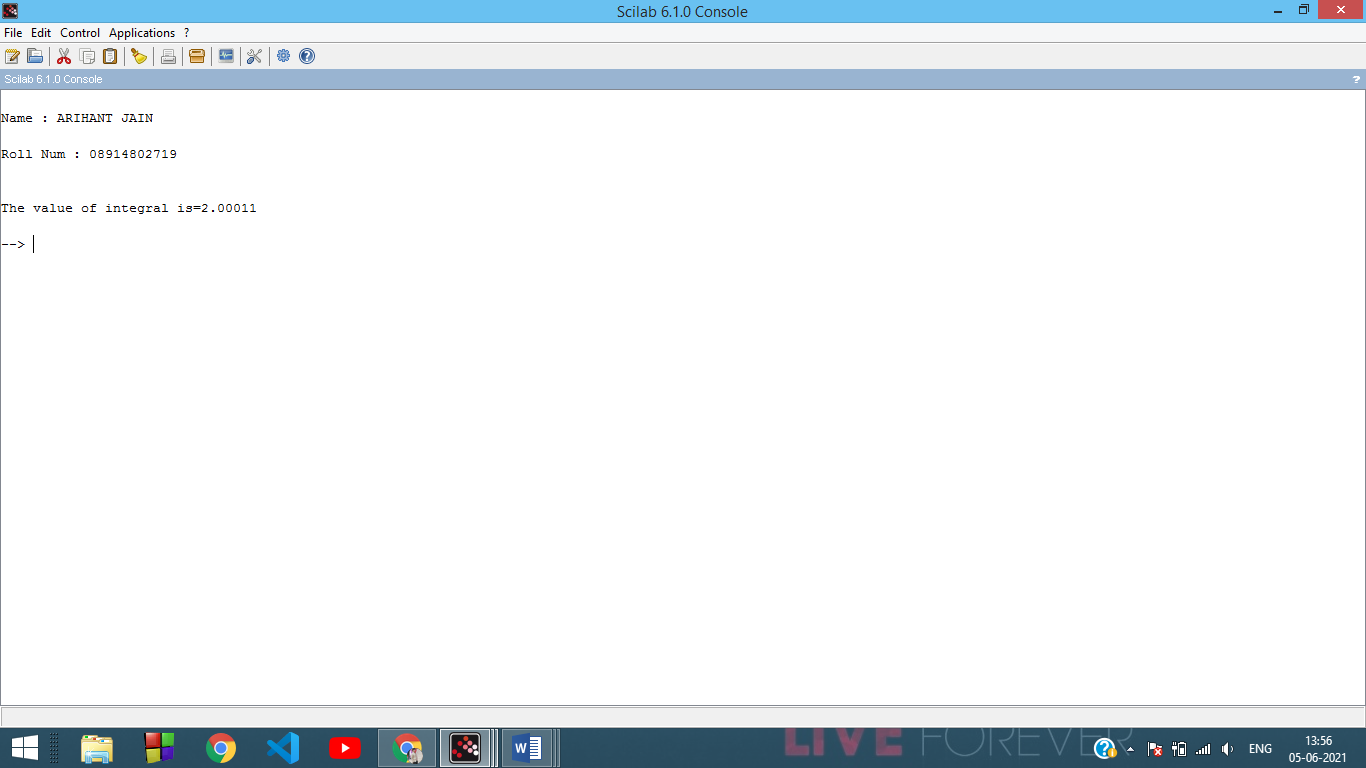
s=s+f(x0+(i-1)\*h)+4\*f(x0+i\*h)+f(x0+(i+1)\*h);

end

integral=(h\*s)/3;

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

***OUTPUT:***



**#Code**

***( using Simpson's three eighth rule)***

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

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

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

x0=0; xn=6;

printf('Integration of y=2/(1+x^2) from 0 to 6');

n=6; *//n | 3*

h=(xn-x0)/n;

s=0;

for i=1:3:n

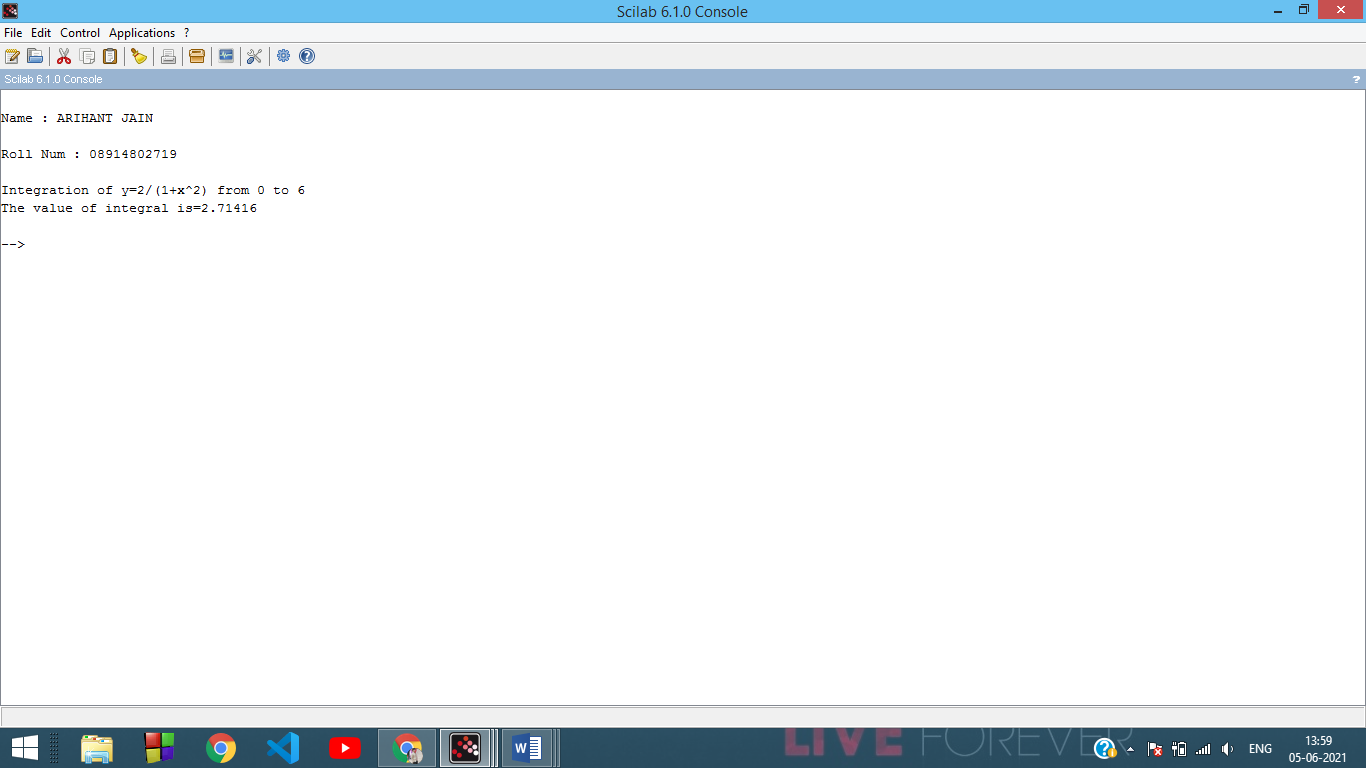
s=s+f(x0+(i-1)\*h)+3\*f(x0+i\*h)+3\*f(x0+(i+1)\*h)+f(x0+(i+2)\*h);

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

**#Code**

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

function **ydot**=fun(**t**, **y**)

**ydot**=**y**^3-**y**\*sin(**t**)+cos(**t**)

endfunction

y0=0;

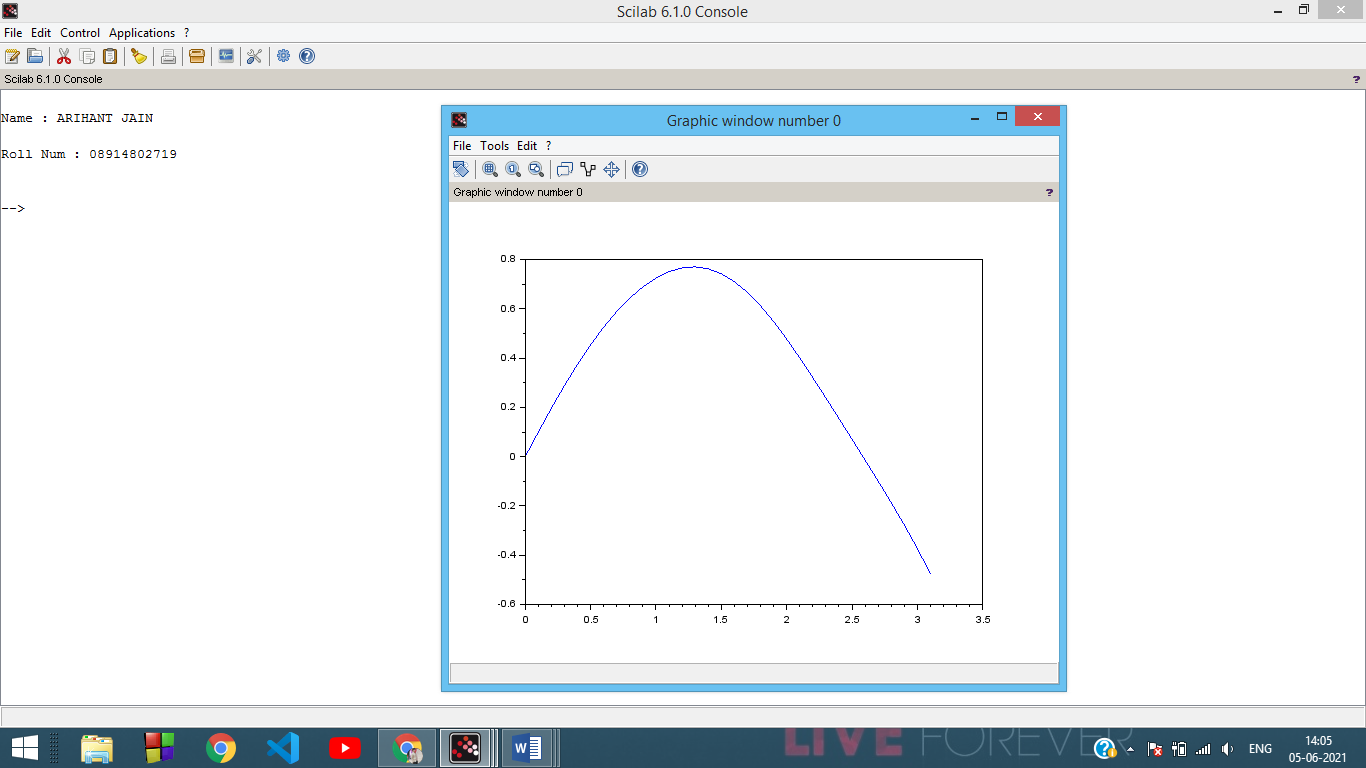
t0=0;

t=0:0.1:%pi;

y = ode(y0,t0,t,fun);

plot(t,y)

***OUTPUT:***



**Experiment - 10**

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

order

**#Code**

clc;

clear ;

d= input("Name : ARIHANT JAIN")

f= input("Roll Num : 08914802719")

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

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

x0=0; y0=1;

xn=0.5;

h=0.1;

x=x0;

y=y0;

lst\_x = []

lst\_y = []

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;

lst\_x($+1) = x;

lst\_y($+1) = y;

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

end

plot2d(lst\_x,lst\_y)

***OUTPUT:***

