**PRACTICAL NO – 4(B)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name :

Roll No :

Aim : Program for Numerical Integration using Simpson’s 1/3rd Rule

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function [**i**]=simpsons13(**a**, **b**, **n**, **f**)

h=(**b**-**a**)/**n**;

x=(**a**:h:**b**);

y=f(x);

m=length(y);

**i**=y(1)+y(m);

for j=2:m-1

if((modulo(j,2)==0)) then

**i**=**i**+4\*y(j);

else

**i**=**i**+2\*y(j);

end

end

**i**=h\***i**/3;

return(**i**);

endfunction

**OUTPUT:**

*-->deff('[y]=f(x)','y=exp(x)-1');*

*-->simpsons13(0,4,8,f)*

*ans =*

*49.616221*