**Assignment: - 07/ Numerical methods (Root finding, numerical differentiation, numerical integration)**

**Write C programs to implement Trapezoidal Rule or Simpson’s 1/3 Rule or Weddle’s Rule to evaluate the integral . *Ans: 0.7854***

#include<stdio.h>

#include<math.h>

float f(float x)

{

return 1/(1+x\*x);

}

main()

{ int n,i;

float s1=0,s2=0,sum,a,b,h;

printf("Enter the values of lower limit & upper limit: ");

scanf("%f%f",&a,&b);

printf("Enter the number of intervals ");

scanf("%d",&n);

h=(b-a)/n;

if(n%2==0)

{for(i=1;i<n;i++)

{if(i%2==0)

s1=s1+f(a+i\*h);

else

s2=s2+f(a+i\*h);}

sum=h\*(f(a)+f(b)+4\*s2+2\*s1)/3;

printf("The value is=%.4f",sum);

}

else

printf("The rule is not applicable");

}

Output

Enter the values of lower limit & upper limit: 0

1

Enter the number of intervals 24

The value is=0.7854

--------------------------------

Process exited after 4.648 seconds with return value 0

Press any key to continue . . .

**Write C programs for Regula-Falsi Method or Newton-Raphson Method to find a real root of the equation: , correct upto 3 decimal places. *Ans: 2.279***

#include<stdio.h>

#include<math.h>

float f(float x)

{

return (x\*x\*x-3\*x-5);}

float d(float x)

{

return (3\*x\*x-3);}

main()

{ float x0,x1,err=0.0001;

printf("Enter the initial guess: ");

scanf("%f",&x0);

if(d(x0)!=0)

{do

{x1=x0;

x0=x0-f(x0)/d(x0);}

while(fabs(x1-x0)>=err);

printf("Root=%.3f",x0);}

}

Output

Enter the initial guess: 3

Root=2.279

--------------------------------

Process exited after 2.873 seconds with return value 0

Press any key to continue . . .