EXPERIMENT NO.: 01

1.1 Objective: To find the root of non-linear equation by Bisection method using C/C++.

1.2 Theory: The bisection method is a method for finding roots of an equation f(x) = 0, given that a root occurs in the interval [a , b]. First the midpoint c=a+ (b-a)/2 of the interval is calculated. Then the function is evaluated at this midpoint. There are three cases:

- f(c) is zero. Here c is the root.
- f(b) and f(c) have opposite signs. In this case the root lies in the interval [b , c].
- f(b) and f(c) have the same signs. In this case the roots lies in the interval [a , c].

1.3 Procedure/Code:

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
float f(float x)
{
   return x*x*x-11*x-9;
}
void main()
{
            float a,b,x,x1,e;
              int I;
      clrscr();
           cout<<"\n Enter the values of a & b: \n";
           cin>>a>>b;
             cout<<"\n Enter error allowed: ";
           cin>>e;
```

```
        I:
    If (f(a)*f(b)>=0)
    {
        cout<<"\nEnter values of a & b again :\
        n"; cin>>a>>b; goto I;
    }
    for(i=0;i<50;i++)
    {
        x=(a+b)/2;
        cout<<"\nValue of x after "<<i+3<<"iteration s : "<<x;
        if(f(a)*f(x)<0) b=x;
        else
            a=x;
        x1=(a+b)/2; if(fabs(x-x1)<=e)
            break;

    }

    cout<<"\nRoot of equation is : "<<x;

    getch();

}
```

1.4 Output: Given equation is x^3 -11x-9=0.

Enter values of a & b :
3
4
Enter error allowed : 0.00001

Value of x after 1 iteration is : 3.5 Value of x after 2 iteration is : 3.75 Value of x after 3 iteration is : 3.625 Value of x after 4 iteration is : 3.6875 Value of x after 5 iteration is : 3.65625 Value of x after 6 iteration is : 3.671875 Value of x after 7 iteration is : 3.664063 Value of x after 8 iteration is : 3.667969 Value of x after 9 iteration is : 3.666016 Value of x after 10 iteration is : 3.666992 Value of x after 11 iteration is : 3.66748 Value of x after 12 iteration is : 3.667725 Value of x after 13 iteration is : 3.667847 Value of x after 14 iteration is : 3.667908 Value of x after 15 iteration is : 3.667938 Value of x after 16 iteration is : 3.667923 Root of equation is : 3.667923

EXPERIMENT NO.: 02

2.1 Objective: To find the root of a non-linear equation by Newton-Raphson Method using C/C++.

2.2 Theory: The Newton-Raphson method is a method for finding roots of an equation f(x)=0 using the formula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

2.3 Procedure/Code:

```
#include<iostream.h>
  #include<conio.h>
#include<math.h>
float f(float a)
{
        float x=((a*a*a)-(4*a)-9);
return(x);
}
float fd(float a)
{
        float x=((3*a*a)-4);
return(x);
}
void main()
{
    clrscr();
        float a,b,fx,fdx,e,itea,k=0;
cout<<"Enter first value";
    cin>>a;
        cout<<"Enter the maximum number of iteration";
cin>>itea;
```

```
cout<<"Maximum acceptable error";
cin>>e;
    for(int i=1;i<itea;i++)
    {
        fx=f(a);
  fdx=fd(a);
b=(a)-(fx/fdx);
fx=f(b);
        if(fabs(fx)<e)
          {
    k++;
  break;
        }
        else
          {
    a=b;
        }
        cout<<"\n Value after"<<i<<" iteration "<<b;
    }
    if(k==0)
    {
        cout<<"Number of iteration is less than required";
      }
getch();
  }
```

2.4 Output:

```
Enter  first value  2
Enter the maximum number of iteration 3
Maximum acceptable error  .0001



Value after 1 iteration  3.125
Value after 2 iteration  2.76853
Value after 3 iteration  2.700196
```

EXPERIMENT NO.: 03

3.1 Objective: To evaluate definite integral by Trapezoidal rule using C/C++.

3.2 Theory:

$$\int_{xo}^{xn} y\ dx = \frac{h}{2}[(y_o + y_n) + 2(y_1 + y_2 + y_3 + \ldots\ldots\ldots\ldots + y_{n-1})]$$

3.3 Procedure/Code:

```
#include<stdio.h>
      #include<conio.h>
#include<math.h>
float y(float x)
{
         return1/(1+x*x);
}
main()
{
    float x0,xn,h,s;
    int I,n;
    clrscr();
    printf("Enter x0, xn, number of subintervals");
    scanf("%f%f%d", &x0, &xn, &n);        h=(xn-x0)/n;
    s=y(x0)+y(xn);       for(i=1;i<=n-1;i++)
    s+=2*y(x0+i*h);
    printf("Value of integral is %6.4f\n",(h/2)*s);
    getch();
}
```

3.4 Result:

```
Enter  x0, xn, number of subintervals
0
6
6
Value of integral is 1.4108
```

EXPERIMENT NO.: 04

4.1 Objective: To evaluate definite integral by Simpson's one-third rule using C/C++.

4.2 Theory: $\int_{xo}^{xn} y\, dx = \frac{h}{3}[(y_o + y_n) + 4(y_1 + y_3 + \ldots + y_{n-1}) + 2(y_2 + y_4 + \ldots + y_{n-2})]$

4.3 Procedure/Code:

```
#include<stdio.h>
  #include<conio.h>
#include<math.h>
float y(float x)
{
      return1/(1+x*x);
}
main()
{
        float x0,xn,h,s;  int I,n;  clrscr();
    printf("Enter x0, xn, number of subintervals");
    scanf("%f%f%d", &x0, &xn, &n);  h=(xn-x0)/n;
    s=y(x0)+y(xn)+4*(x0+h);  for(i=3;i<=n-1;i+=2)
    s+=4*y(x0+i*h)+2*y(x0+(i-1)*h);  printf("Value
    of integral is %6.4f\n",(h/3)*s);  getch();
 }
```

4.4 Result:

```
Enter  x0, xn, number of subintervals
0
6
6
Value of integral is 1.4108.
```