Lab 2

TITLE:- Implementation of false position method using C.

Theory:-

It is another bracketing method to find the solution of non-linear equation. It provides the solution in lesser number of steps or iteration in comparision to bisection method. To overcome the time and space complexity false position method was introduced.

Algorithm:-

i) Define function $f(x)$

ii) Take initial guess x_0 & x_1 and error 'e'.

iii) check $f(x_0) * f(x_1) > 0$ our initial guess is wrong otherwise continue.

iv) Calculate

$$x_2 = x_0 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} \times f(x_0)$$

v) if $f(x_0) \times f(x_2) > 0$

assign $x_0 = x_2$

else,

$x_1 = x_2$

vi) perform till stopping criteria

vii) print the root as x_2

viii) stop.

Source code:-

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
    float f(float x) {
        return x*x*x-4*x-9;
        }
    int main(){
```

```
float a,b,c,e=0.001;
printf("enter the initial guesses a and b");
scanf("%f %f",&a,&b);
if(f(a)*f(b)>0){
printf("Ouv initial guess is wrong");
}
else{
   do{
   c = a-(b-a)*f(a)/(f(b)-f(a));
    if(f(a)*f(c)>0){
        a=c;
      }
      else{
        b=c
        }
  printf("%f if %f if %f if %f if %f",a,b,f(a),f(b),f(c));
   {
while(fabs(f(c))>c);
{
printf("the root of given eqn is %f",c);
}
```

CONCLUSION:-

The root of given eqn $f(x) = x^3-4x-9$ is 2.70025 and hence we became familiar with the implementation of false position method using C.

Lab:- 3

TITLE :- Implementation of Secant method using C.

Theory:

This another method to find the solution of non-linear equation using non-bracketing technique. It provides optimized result in less no of iteration.

Algorithm:-

i) Define function f(x) and tolerance (e)

ii) input initial guesses x_0 & x_1

iii) Compute $x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} \times f(x_1)$

assign

$x_0 = x_1$

and

$x_1 = x_2$

iv) perform till stopping criteria

v) print the root x_2

vi) stop.

Formula:-

$$x_2 = x_1 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} \times f(x_1)$$

Source code:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
float f(float x){
    return x*x*x-2*x-9;
    }
int main(){
float a,b,c, e=0.001;
printf("Enter the initial guesses a and b");
}
```

```
else{
 do{
  c = b-(b-a) * f(b)/(f(b) - f(a));
  if(f(a)* f(c)>0){
    a = c;
  }
  else{
    b = c;
  }
printf("%f\t%f\t%f\t%f\t%f\n",a,b,f(a),f(b),c,
                                   f(c));
 }
 while (fabs(f(c)>e);
 }
 printf("The root of the given equation is %f", c);
 }
```

CONCLUSION:

The root of given ~~eqn~~ equation is 2.70025

hence we became familiar with implementation of secant method using c.

Lab-4: Newton Raphsion Method

Theory:-

The Newton-Raphson method, also known as Newton's method, is an iterative numberical technique for finding the root of a real-valued function.

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

C code for Newtons Raphson

```
#include<stdio.h>
#include<math.h>
float f(float x){
  return x*x*x-4*x-9;
}
int main(){
float a, c; e=0.0001;
int main(){ int step=1;
printf("Enter the initial guesses a");
scanf("%f", &a);
if(fdash(a)==0){
printf("Invalid eq\n");
}else{
  printf("|%-4s|%-8s|%-8s|%-8s|\n","step","a","c","f(c)");
Print("|----|----|----|----|\n");
do{
 c=a-f(a)/fdash(a);
printf("|%-4d|%-8.4f|%-8.4f|%-8.4|\n", step,a,c,f(c));
a=c;
step++;
}
while(fabs(f(c))>e);
printf("|----|----|----|----|\n");
```

```
printf("final Answer: c = %f\n",c);
}
return 0;
}
```

Conclusion :-

Hence the value of x is found to be 2.6570 using newton raphson method.