Problem 01:

Determination of a root correct to three decimal places for an equation using Bisection Method and Flase-Position Method.

Theory:

Bisection Method is based on that if a function continuous between a and b ,and f(a) and f(b) are of opposite signs,then there exits at least one root between a and b if f(a) is negative and f(b) is positive, then the root lies between a and b ,and let its approximate value be given by Xr=(a+b)/2. If (Xr)=0, we can say Xr is a root of the equation. Otherwise root is between Xr and b or Xr and a depending on whether f(Xr) positive or negative. False position method is almost same as bisection method. This method is also known as regula_falsi method.We choose two points a and b such that f(a) and f(b) are of opposite signs.Hence, a root must lie in between this points.Now, the equations of the chord joining the two points [a,f(a)] and [b,f(b)] is given by

$$Xr = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

Code:

```
#include<bits/stdc++.h>
#include<conio.h>
using namespace std;

double a,b,c,d;

double valueCheck(double x)
{
   float ans=a*pow(x,3)+b*pow(x,2)+c*x+d;
   return ans;
}

int main(){
   cout<<"The Equation of f(x) : ax^3+bx^2+cx+d=0\n";
   cout<<"Enter the value of a,b,c and d:";
   cin>>a>>b>>c>>d;

   int temp=-1;

   while(temp!=3){
      system("cls");
      double i=0,x=0,y=0,prev=0,root;
      int p=0,q=0;
      cout<<"1.Bisection Method\n2.False Position Method\n3.Exit\n";
      cout<<"Enter the method:";
      cin>>temp;
      while(true){
```

```
    if(valueCheck(i)<0&&p==0)
    {
        x=i;
        p=1;
    }
    else if(valueCheck(i)>0&&q==0){
        y=i;
        q=1;
    }
    double j=0-i;
    if(valueCheck(j)<0&&p==0){
        x=j;
        p=1;
    }
    else if(valueCheck(j)>0&&q==0)
    {
        y=j;
        q=1;
    }
    if(p!=0&&q!=0){
        break;
    }
    i++;
}
double Xr;
int k=1;
while(true){
    if(temp==1){
        Xr=(x+y)/2;
    }else{
        double f=valueCheck(x);
        double g=valueCheck(y);
        Xr=(x*g-y*f)/(g-f);
    }
    if(fabs(Xr-prev)<=0.0001)
    {
        break;
    }
    if(valueCheck(Xr)<0){
         x=Xr;
    }
    else if(valueCheck(Xr)>0){
        y=Xr;
    }
    else{
```

```
            break;
          }
          prev=Xr;
          cout<<k<<"|"<<x<<"|"<<y<<"|"<<Xr<<"|"<<
          valueCheck(Xr)<<"|"<<fabs(valueCheck(Xr))<<endl;
          cout<<"________________________________________"<<endl;
          k++;
        }
        cout<<"So root is:"<<Xr<<endl;
        cout<<endl;
        getch();
    }

    return 0;
}
```

Result/Output:

```
The Equation of f(x) : ax^3+bx^2+cx+d=0
Enter the value of a,b,c and d:1 0 -1 -1
1.Bisection Method
2.False Position Method
3.Exit
Enter the method:1
1|1|2|1|-1|1
________________________________________
2|1|1.5|1.5|0.875|0.875
________________________________________
3|1.25|1.5|1.25|-0.296875|0.296875
________________________________________
4|1.25|1.375|1.375|0.224609|0.224609
________________________________________
5|1.3125|1.375|1.3125|-0.0515137|0.0515137
________________________________________
6|1.3125|1.34375|1.34375|0.0826111|0.0826111
________________________________________
7|1.3125|1.32812|1.32812|0.014576|0.014576
________________________________________
8|1.32031|1.32812|1.32031|-0.0187106|0.0187106
________________________________________
9|1.32422|1.32812|1.32422|-0.00212795|0.00212795
________________________________________
10|1.32422|1.32617|1.32617|0.00620883|0.00620883
________________________________________
11|1.32422|1.3252|1.3252|0.00203665|0.00203665
________________________________________
```

12|1.32471|1.3252|1.32471|-4.65949e-005|4.65949e-005

13|1.32471|1.32495|1.32495|0.000994791|0.000994791

14|1.32471|1.32483|1.32483|0.000474039|0.000474039

So root is:1.32477

1.Bisection Method
2.False Position Method
3.Exit
Enter the method:2
1|0.333333|2|0.333333|-1.2963|1.2963

2|0.676471|2|0.676471|-1.36691|1.36691

3|0.960619|2|0.960619|-1.07417|1.07417

4|1.14443|2|1.14443|-0.645561|0.645561

5|1.24226|2|1.24226|-0.325196|0.325196

6|1.28853|2|1.28853|-0.149163|0.149163

7|1.30914|2|1.30914|-0.0654644|0.0654644

8|1.31807|2|1.31807|-0.028173|0.028173

9|1.32189|2|1.32189|-0.012022|0.012022

10|1.32352|2|1.32352|-0.00511143|0.00511143

11|1.32421|2|1.32421|-0.00216989|0.00216989

12|1.3245|2|1.3245|-0.000920553|0.000920553

13|1.32463|2|1.32463|-0.000390426|0.000390426

So root is:1.32468

Conclusion :

The code successfully implements both the Bisection and False-Position Methods to compute the root of a cubic equation. By iterating through the process and refining the root estimate, it guarantees finding the root accurate to three decimal places, provided that the function f(x)f(x)f(x) is continuous, and f(a)f(a)f(a) and f(b)f(b)f(b) have opposite signs.