When implementing the method on a computer, there can be problems with finite precision, so there are additional convergence tests or limits to the number of iterations. Although f is continuous, finite precession may preclude a function value ever being zero. For f(x)=x-pi, there will never be a finite representation of X that gives zero. Floating point representations also have limited precision, so at some point the midpoint of [a,b] will be either a or b.

**Source code:**

#include<bits/stdc++.h>

using namespace std;

double f(double x);

double f(double x)

{

double p=pow(x,3)-x-11;

return p;

}

int main()

{

double p,q,r,t,fp,fq,fr;

p:cout<<"Enter any initial guesses : \np =";

cin>>p;

cout<<"\nq =";

cin>>q;

cout<<"\n Enter the degree of accuracy desired "<<endl;

cin>>t;

if(f(p)\*f(q)>0)

{

cout<<"Please enter a different initial guesses : "<<endl;

goto p;

}

else

{

while(fabs(p-q)>=t)

{

r=(p+q)/2;

fp=f(p);

fq=f(q);

fr=f(r);

cout<<" p ="<<p<<" "<<" q ="<<q<<" "<<" r ="<<r<<" fr ="<<fr<<endl;

if(fr==0)

{

cout<<"The root of the eqn is "<<r;

}

if(fp\*fr>0)

p=r;

else if (fp\*fr<0)

q=r;

}

}

cout<<"The root of the eqn is "<<r;

return 0;

}