## Problem Define:

Determination of a root correct to three decimal places for an equation using iteration method

## Theory:

The Iteration Method or Fixed Point Iteration Method is a numerical method used to solve equations of the form f(x)=0f. The basic idea of this method is to express the function f(x) as a rewrite new function g(x) such that:

$$x=g(x)$$

The root-finding process involves starting with an initial guess x0 and then iteratively computing new approximations to the root using the formula:

$X_{n+1}=g(X_n)$

Where:

- x0 is the initial guess.
- Xn+1 is the new approximation obtained from the previous value Xn.
- g(x) is derived from rearranging the original equation f(x)=0.

**Convergence Criteria**:

For the iteration method to converge, the function g(x) should satisfy certain conditions:

- g(x) should be continuous in the interval containing the root.
- The derivative)g′(x) should satisfy |g′(x)|<1 in the interval around the root, ensuring the convergence of the iterative process.

The number of iterations required to reach the desired accuracy depends on:

- The initial guess x0.
- The function g(x) and its derivative.
- The tolerance ϵ\epsilonϵ (tolerable error) specified.

## Code:

```
#include<bits/stdc++.h>
/* Define function f(x) which is to be solved */
#define f(x)    sin(x)-10*x+10
/* Rewrite f(x) as x=g(x) and define g(x) here */
#define g(x)    1+(sin(x)/10)
using namespace std;

int main(){
    int step=1,N;
    double x0,x1,e;
    cout<<setprecision(3)<<fixed;
    //inputs
    cout<<"Enter inital guess: ";
    cin>>x0;
    cout<<"Enter tolerable error: ";
    cin>>e;
```

```
    cout<<"Enter maximum iteration: ";
    cin>>N;

    //Implementing Fixed Point Iteration
    cout<<"\n******************************"<<endl;
    cout<<"Fixed Point Iteration Method"<<endl;
    cout<<"*******************************"<<endl;

    do{
        x1=g(x0);
        cout<<"Iteration-"<<step<<":\t x1 = "<<setw(10)<<x1<<" and f(x1) = 
"<<setw(10)<<f(x1)<<endl;
        step++;
        if(step>N){
            cout<<"Not Convergent.";
            break;
        }
        x0=x1;
    }while(fabs(f(x1))>e);

    cout<<"\nRoot is : "<<x1;
    return 0;
}
```

## Output:

Enter inital guess: 1
Enter tolerable error: 0.001
Enter maximum iteration: 10

******************************
Fixed Point Iteration Method
********************************
Iteration-1: x1 = 1.084 and f(x1) = 0.042
Iteration-2: x1 = 1.088 and f(x1) = 0.002
Iteration-3: x1 = 1.089 and f(x1) = 0.000

Root is : 1.089

## Conclusion:

The Iteration Method is a simple and effective numerical method for finding the root of a nonlinear equation. However, its success depends on proper selection of the function g(x) and the initial guess x0. When the method converges, it can provide highly accurate solutions. The primary advantages of the Iteration Method include its ease of implementation and low computational cost. However, convergence is not guaranteed unless the function meets the necessary criteria. In practical scenarios, it is essential to analyze the behavior of g(x) and its derivative to ensure convergence.

## Problem Define:

Determination of a root correct to three decimal places for an equation using Newton_Rapshon

## Theory:

The Newton-Raphson Method is a powerful iterative technique used for finding successively better approximations to the roots (or zeroes) of a real-valued function. The method is based on the idea of linear approximation. If you have a function f(x) that is differentiable, the root can be approximated using the tangent line at a current approximation of the root.

**Mathematical Foundation**

The method uses the following formula:

$$xn+1=xn-f(n)/f'(xn)$$

Where:

- xn is the current approximation of the root.
- f(xn) is the function evaluated at xn.
- f′(xn) is the derivative of the function evaluated at xn.
- xn+1 is the next approximation.

Procedure:

- Choose an Initial Guess**:** Start with an initial approximation x0 that is reasonably close to the actual root.
    - Iterate**:** Apply the Newton-Raphson formula to compute the next approximation:

$$xn+1=xn-f(n)/f'(xn)$$

- Convergence Check: Repeat the iteration until the absolute difference between successive approximations is less than a specified tolerance ϵ\epsilonϵ, or until |f(xn+1)|<ϵ

**Convergence Criteria:**

The convergence of the Newton-Raphson method can be rapid, particularly when the initial guess is close to the actual root. However, certain conditions must be met for the method to converge:

- The function f(x) should be continuous and differentiable in the neighborhood of the root.
- The derivative f′(x) should not be zero at the root.
- A good initial guess is essential; if the guess is too far from the root, the method may diverge or converge to a different root.

Code:

```
#include<bits/stdc++.h>
/*Defining equation to be soved.
change this equation to solve another problem.*/
#define f(x)  3*x-cos(x)-1
/* Defining derivative of g(x).
As you change f(x), change this function also.*/
```

```
#define g(x)  3+sin(x)
using namespace std;

int main()
{
    double x0,x1,f0,f1,g0,e;
    int step=1, N;

    //Setting precision and writing floating point
    //values in fixed_point notation.
    cout<<setprecision(4)<<fixed;

    //inputs
    cout<<"Enter initial guess: ";
    cin>>x0;
    cout<<"Enter tolerable error: ";
    cin>>e;
    cout<<"Enter maximum iteration: ";
    cin>>N;

    //Implementing Newton Raphson Method
    cout<<"\n*******************************\n";
    cout<<"Newton Rapshon Method"<<endl;
    cout<<"*******************************"<<endl;

    do{
        f0=f(x0);
        g0=g(x0);

        if(g0==0.0){
            cout<<"Mathmetical Error.";
            break;
        }

        x1=x0-(f0/g0);

        cout<<"Iteration-"<<step<<":\t x = "<<setw(10)<<x1<<" and f(x) =
"<<setw(10)<<f(x1)<<endl;
        x0=x1;

        step++;

        if(step>N){
            cout<<"Not Convergent.";
            break;
```

```
        }
        f1=f(x1);
    }while(fabs(f1)>e);

    cout<<"\nRoot is: "<<x1;
    return 0;
}
```

## Output:

Enter initial guess: 2
Enter tolerable error: 0.0001
Enter maximum iteration: 10

*********************************
Newton Rapshon Method
*********************************
Iteration-1: x = 0.615 and f(x) = 0.027
Iteration-2: x = 0.607 and f(x) = 0.000

Root is: 0.607

## Conclusion:

The Newton-Raphson Method is an efficient numerical technique for finding roots of real-valued functions. Its quadratic convergence rate often leads to very rapid approximation of roots, especially when the initial guess is close to the true root. The method is widely used in various fields, including engineering, physics, and applied mathematics.the method has some limitations: If the derivative f'(x) is zero or very small near the root, the method may fail to converge or may lead to significant errors. The method may diverge if the initial guess is not sufficiently close to the true root.It may also converge to local minima or maxima instead of the desired root if the function has multiple roots. In practical applications, it is essential to analyze the behavior of the function and its derivative and to select an appropriate initial guess. When used appropriately, the Newton-Raphson Method can provide highly accurate solutions to a wide range of problems.