**Write a C program to simulate Bisection Method.**

* ***Source code***

#include<stdio.h>

#include<conio.h>

#include<math.h>

/\*

Defining equation to be solved.

Change this equation to solve another problem.

\*/

#define f(x) cos(x) - x \* exp(x)

main()

{

float x0, x1, x2, f0, f1, f2, e;

int step = 1;

/\* Inputs \*/

up:

printf("\nEnter two initial guesses:\n");

scanf("%f%f", &x0, &x1);

printf("Enter tolerable error:\n");

scanf("%f", &e);

/\* Calculating Functional Value \*/

f0 = f(x0);

f1 = f(x1);

/\* Checking whether given guesses brackets the root or not. \*/

if( f0 \* f1 > 0.0)

{

printf("Incorrect Initial Guesses.\n");

goto up;

}

/\* Implementing Bisection Method \*/

printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");

do

{

x2 = (x0 + x1)/2;

f2 = f(x2);

printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

if( f0 \* f2 < 0)

{

x1 = x2;

f1 = f2;

}

else

{

x0 = x2;

f0 = f2;

}

step = step + 1;

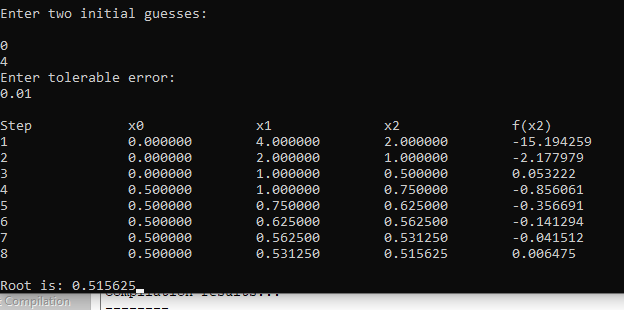
}while(fabs(f2)>e);

printf("\nRoot is: %f", x2);

getch();

}

* ***Output***

******

**Write a C program to simulate False Position.**

* ***Solution***

#include<stdio.h>

#include<conio.h>

#include<math.h>

/\* Defining equation to be solved.

Change this equation to solve another problem. \*/

#define f(x) x\*log10(x) - 1.2

int main()

{

float x0, x1, x2, f0, f1, f2, e;

int step = 1;

/\* Inputs \*/

up:

printf("\nEnter two initial guesses:\n");

scanf("%f%f", &x0, &x1);

printf("Enter tolerable error:\n");

scanf("%f", &e);

/\* Calculating Functional Values \*/

f0 = f(x0);

f1 = f(x1);

/\* Checking whether given guesses brackets the root or not. \*/

if( f0\*f1 > 0.0)

{

printf("Incorrect Initial Guesses.\n");

goto up;

}

/\* Implementing Regula Falsi or False Position Method \*/

printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");

do

{

x2 = x0 - (x0-x1) \* f0/(f0-f1);

f2 = f(x2);

printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

if(f0\*f2 < 0)

{

x1 = x2;

f1 = f2;

}

else

{

x0 = x2;

f0 = f2;

}

step = step + 1;

}while(fabs(f2)>e);

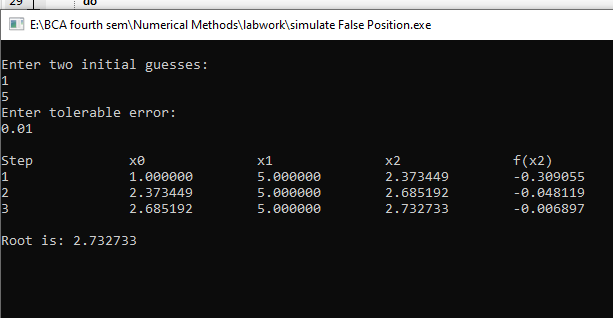
printf("\nRoot is: %f", x2);

getch();

return 0;

}

* ***Output***



**Write a C program to simulate Newton-Raphson.**

* ***Source code***

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<stdlib.h>

/\* Defining equation to be solved.

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)

main()

{

float x0, x1, f0, f1, g0, e;

int step = 1, N;

/\* Inputs \*/

printf("\nEnter initial guess:\n");

scanf("%f", &x0);

printf("Enter tolerable error:\n");

scanf("%f", &e);

printf("Enter maximum iteration:\n");

scanf("%d", &N);

/\* Implementing Newton Raphson Method \*/

printf("\nStep\t\tx0\t\tf(x0)\t\tx1\t\tf(x1)\n");

do

{

g0 = g(x0);

f0 = f(x0);

if(g0 == 0.0)

{

printf("Mathematical Error.");

exit(0);

}

x1 = x0 - f0/g0;

printf("%d\t\t%f\t%f\t%f\t%f\n",step,x0,f0,x1,f1);

x0 = x1;

step = step+1;

if(step > N)

{

printf("Not Convergent.");

exit(0);

}

f1 = f(x1);

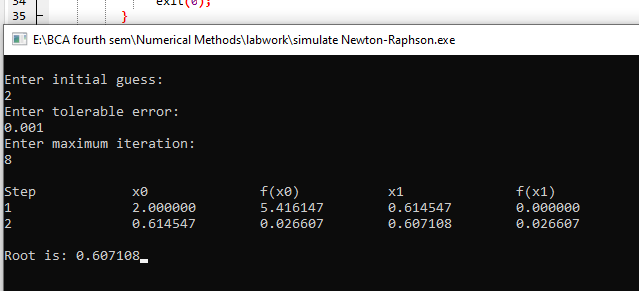
}while(fabs(f1)>e);

printf("\nRoot is: %f", x1);

getch();

}

* ***Output***

******