Source Code:

//Bisection (a) -> x^2 - 4x -10 = 0

#include <stdio.h.>

#include <math.h>

#include <stdlib.h>

float f (float x)

{

return (x\*x - 4\*x - 10);

}

float absolute(float n)

{

return(n < 0 ? -1.0 \* n : n);

}

float Bisection(float a, float b, float e)

{

float c, error, prev\_c = 0;

int iteration = 0, loop = 1;

while (loop == 1)

{

iteration ++;

//bisection

c = (a+b)/2;

printf("\nIteration %d \t|a = %f \t|b = %f \t|c = %f", iteration, a, b, c);

//new interval

if (f(a) \* f(c) < 0){

b = c;

}

else if (f(b) \* f(c) < 0){

a = c;

}

else{ }

error = absolute((c - prev\_c)/c);

if(error <= e){

loop = 0;

printf("\n\nE = %f, in Iteration %d", e, iteration);

}

else if(iteration > 200){

exit(1);

}

prev\_c = c;

}

return c;

}

int main()

{

float result, a, b, e, root, slope, fa, fb;

int i;

printf("\nEnter initial range : ");

scanf("%f%f", &a, &b);

printf("\nEnter acceptable error : ");

scanf("%f", &e);

/\*

Here in the while block below we make use of slope made by the initial

guesses to determine whether to increase b or decrease a.

\*/

while(f(a)\*f(b) > 0)

{

fa = f(a);

fb = f(b);

slope = (fb - fa) / (b - a);

if(slope > 0)

{

(fa > 0 && fb > 0)? a-- : b++ ;

}

else if(slope < 0)

{

(fa > 0 && fb > 0)? b++ : a-- ;

}

else{ // deals with infinite slope or zero slope cases.

b++;

}

}

printf("\nOperting with initial assumptions : x0 = %f, x1 = %f .", a, b);

if(f(a) == 0){

root = a;

}

else if(f(b) == 0){

root = b;

}

else {

root = Bisection(a, b, e);

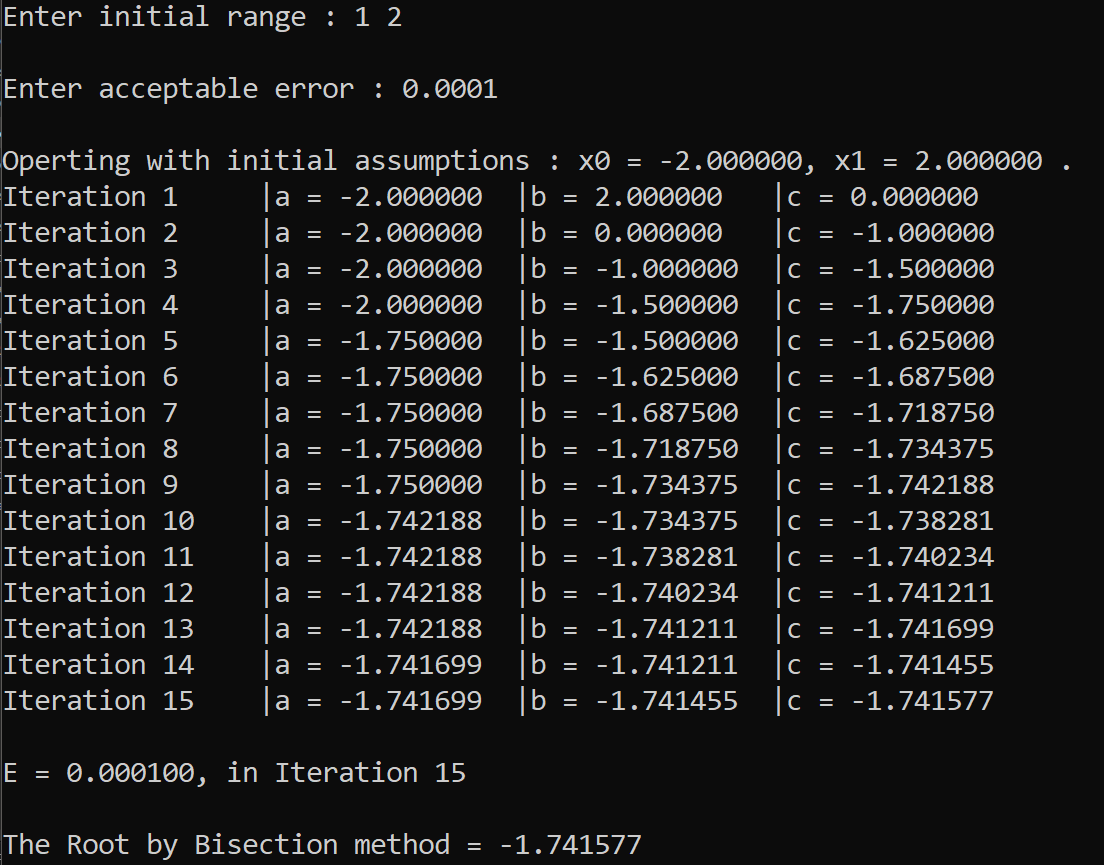
}

printf("\n\nThe Root by Bisection method = %f", root);

return 0;

}

Output:



Source Code:

//Bisection (b) -> 4sin(x) = e^x

#include <stdio.h.>

#include <stdlib.h>

#include <math.h>

float f (float x)

{

return (4\*sin(x) - exp(x));

}

float absolute(float n)

{

return(n < 0 ? -1.0 \* n : n);

}

float Bisection(float a, float b, float e)

{

float c, error, prev\_c = 0;

int iteration = 0, loop = 1;

while (loop == 1)

{

iteration ++;

//bisection

c = (a+b)/2;

printf("\nIteration %d \t|a = %f \t|b = %f \t|c = %f", iteration, a, b, c);

//new interval

if (f(a) \* f(c) < 0){

b = c;

}

else if (f(b) \* f(c) < 0){

a = c;

}

else{ }

error = absolute((b - a)/b);

if(error <= e){

loop = 0;

printf("\n\nE = %f, in Iteration %d", e, iteration);

}

else if(iteration > 200){

exit(1);

}

prev\_c = c;

}

return c;

}

int main()

{

float result, a, b, e, root, slope, fa, fb;

int i;

printf("\nEnter initial range : ");

scanf("%f%f", &a, &b);

printf("\nEnter acceptable error : ");

scanf("%f", &e);

/\*

Here in the while block below we make use of slope made by the initial

guesses to determine whether to increase b or decrease a.

\*/

while(f(a)\*f(b) > 0)

{

fa = f(a);

fb = f(b);

slope = (fb - fa) / (b - a);

if(slope > 0)

{

(fa > 0 && fb > 0)? a-- : b++ ;

}

else if(slope < 0)

{

(fa > 0 && fb > 0)? b++ : a-- ;

}

else{ // deals with infinite slope or zero slope cases.

b++;

}

}

printf("\nOperting with initial assumptions : x0 = %f, x1 = %f .", a, b);

if(f(a) == 0){

root = a;

}

else if(f(b) == 0){

root = b;

}

else {

root = Bisection(a, b, e);

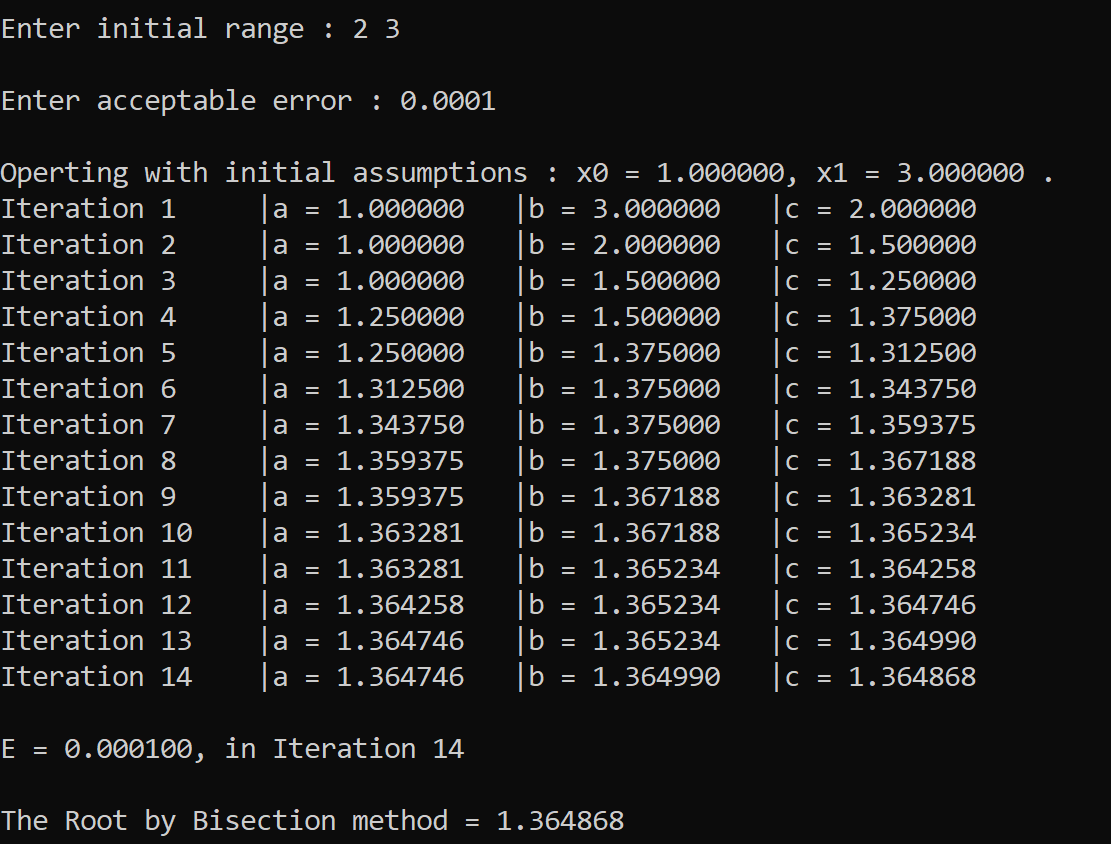
}

printf("\n\nThe Root by Bisection method = %f", root);

return 0;

}

Output:



Source Code:

//False Position (a) -> x^2 - x -2 = 0

#include <stdio.h.>

#include <stdlib.h>

#include <math.h>

float f (float x)

{

return (x\*x - x - 2);

}

float absolute(float n)

{

return(n < 0 ? -1.0 \* n : n);

}

float FalsePos(float a, float b, float e)

{

float c, error, prev\_c = 0;

int iteration = 0, loop = 1;

while (loop == 1)

{

iteration ++;

//bisection

c = a - ((f(a)\*(b-a))/(f(b) - f(a)));

printf("\nIteration %d \t|a = %f \t|b = %f \t|c = %f ", iteration, a, b, c);

//new interval

if (f(a) \* f(c) < 0){

b = c;

}

else if (f(b) \* f(c) < 0){

a = c;

}

else{ }

error = absolute((c - prev\_c)/c);

printf("\t|E = %f ", error);

if(error <= e){

loop = 0;

printf("\n\nE = %f, in Iteration %d", e, iteration);

}

else if(iteration > 200){

exit(1);

}

prev\_c = c;

}

return c;

}

int main()

{

float result, a, b, e, root, slope, fa, fb;

int i;

printf("\nEnter initial range : ");

scanf("%f%f", &a, &b);

printf("\nEnter acceptable error : ");

scanf("%f", &e);

/\*

Here in the while block below we make use of slope made by the initial

guesses to determine whether to increase b or decrease a.

\*/

while(f(a)\*f(b) > 0)

{

fa = f(a);

fb = f(b);

slope = (fb - fa) / (b - a);

if(slope > 0)

{

(fa > 0 && fb > 0)? a-- : b++ ;

}

else if(slope < 0)

{

(fa > 0 && fb > 0)? b++ : a-- ;

}

else{ // deals with infinite slope or zero slope cases.

b++;

}

}

printf("\nOperting with initial assumptions : x0 = %f, x1 = %f .", a, b);

if(f(a) == 0){

root = a;

}

else if(f(b) == 0){

root = b;

}

else {

root = FalsePos(a, b, e);

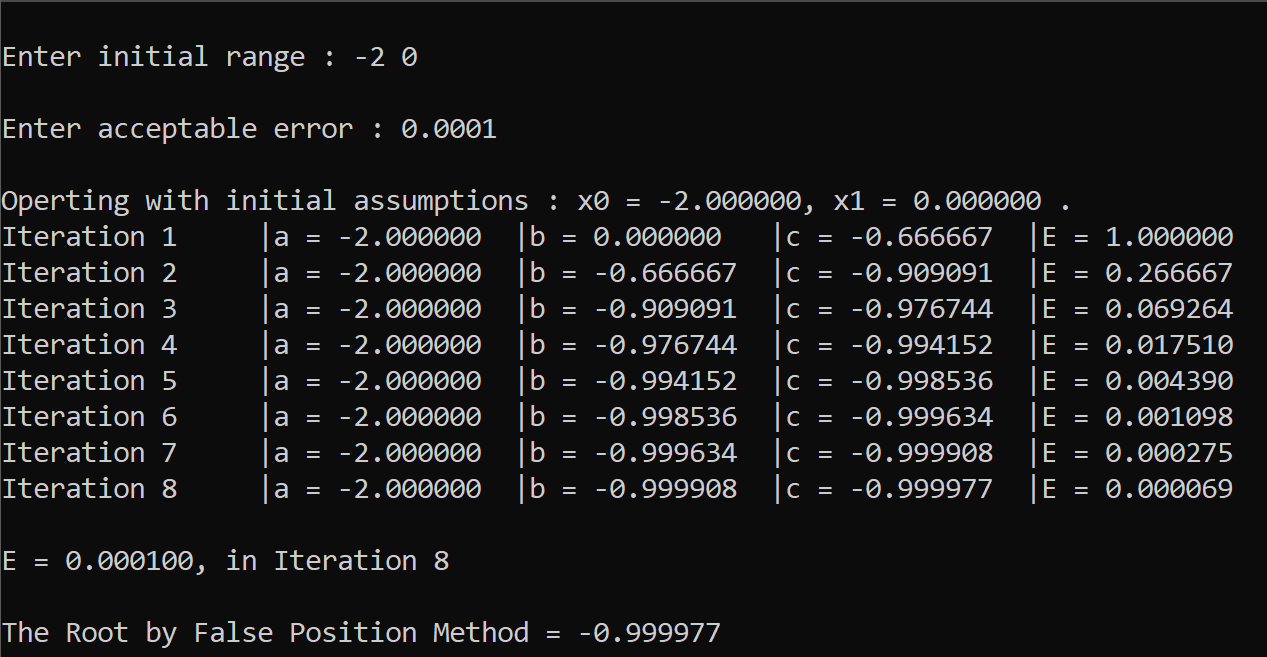
}

printf("\n\nThe Root by False Position Method = %f", root);

return 0;

}

Output:



Source Code:

//False Position (b) -> x\*e^x -2 = 0

#include <stdio.h.>

#include <stdlib.h>

#include <math.h>

float f (float x)

{

return ( x \* exp(x) - 2);

}

float absolute(float n)

{

return(n < 0 ? -1.0 \* n : n);

}

float FalsePos(float a, float b, float e)

{

float c, error, prev\_c = 0;

int iteration = 0, loop = 1;

while (loop == 1)

{

iteration ++;

//bisection

c = a - ((f(a)\*(b-a))/(f(b) - f(a)));

printf("\nIteration %d \t|a = %f \t|b = %f \t|c = %f ", iteration, a, b, c);

//new interval

if (f(a) \* f(c) < 0){

b = c;

}

else if (f(b) \* f(c) < 0){

a = c;

}

else{ }

error = absolute((c - prev\_c)/c);

printf("\t|E = %f ", error);

if(error <= e){

loop = 0;

printf("\n\nE = %f, in Iteration %d", e, iteration);

}

else if(iteration > 200){

exit(1);

}

prev\_c = c;

}

return c;

}

int main()

{

float result, a, b, e, root, slope, fa, fb;

int i;

printf("\nEnter initial range : ");

scanf("%f%f", &a, &b);

printf("\nEnter acceptable error : ");

scanf("%f", &e);

/\*

Here in the while block below we make use of slope made by the initial

guesses to determine whether to increase b or decrease a.

\*/

while(f(a)\*f(b) > 0)

{

fa = f(a);

fb = f(b);

slope = (fb - fa) / (b - a);

if(slope > 0)

{

(fa > 0 && fb > 0)? a-- : b++ ;

}

else if(slope < 0)

{

(fa > 0 && fb > 0)? b++ : a-- ;

}

else{ // deals with infinite slope or zero slope cases.

b++;

}

}

printf("\nOperting with initial assumptions : x0 = %f, x1 = %f .", a, b);

if(f(a) == 0){

root = a;

}

else if(f(b) == 0){

root = b;

}

else {

root = FalsePos(a, b, e);

}

printf("\n\nThe Root by False Position Method = %f", root);

return 0;

}

Output:

