

```
/*
Finding the root of an equation f(x) by using regular falsi method..

Regula Falsi method assumes that function as linear.. even though these methods are needed
only when f(x) is not linear, and usually work well anyway.

The ratio of change in x, to the resulting change in y is :
      (X2 - X1)/(Y2 - Y1)
Because y, most recently, is Y2, and we want y to be 0, then the change that we want in y is
0 - Y2 which is equal to -Y2.

So, given that desired change in y, and given the expected ratio of change in x to change in
y, then the best estimate for the right x-value nothing but the best estimation for the root
is :
The latest value x plus the product of the desired change in y and the expected ratio of
change in x to change in y:

X3 = X2 + -(Y2) (X2-X1)/(Y2-Y1)
this formula for X3 is adequate, but can be put in a more practical form

By rearranging the above equation we get

X3 = (X1*Y2 - X2Y1)/(Y2 - Y1)

Gradually the equation converges to a best approximation to the root of the equation f(x)
which we take it as our required root.
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

//Function prototypes
float f(float x);
float falsi(float a, float b);

int main()
{
    float a, b; //Interval boundaries
    float root; //Root of the function f

    //Getting interval from the user
    printf("Enter a: ");
    scanf("%f", &a);
    printf("Enter b: ");
    scanf("%f", &b);

    root = falsi(a, b); //Calling falsi function
    printf("Root of the given equation: %f\n", root);

    exit (0);
}

float falsi(float a, float b)
{
    float c, x, y, root;

    //Loop infinitely
    while (1)
    {
        //Updating values based on the conditions satisfied
        if(f(a)*f(b) < 0)
        {
            root = (a*f(b) - b*f(a))/(f(b) - f(a));
            c = root;
        }
    }
}
```

```
if(f(b)*f(c) < 0)
{
    root = b - (((c-b)/(f(c)-f(b)))*f(b)); // Function Equation
    a = b;
    b = c;
    c = root;
}
else
{
    root = a - (((c-a)/(f(c)-f(a)))*f(a)); // Function Equation
    a = b;
    b = c;
    c = root;
}

//Round off a and b upto 5 decimal accuracy
x = round(b*100000)/100000;
y = round(root*100000)/100000;

//Checking the convergence of the equation
if(x == y)
{
    return root;
}
}

float f(float x)
{
    float ans;

    //Calculating the function value
    ans = 3*x - cosf(x) - 1;

    //Checking of the result is NAN
    if(ans != ans)
    {
        printf("Cannot proceed further..Try changing the interval\n");
        exit (2);
    }
    return ans;
}
```