Bisection Method

Example 1: *// Find a real root of equation f(x)=x3-4x-9=0 using bisection method correct to three decimal places*

*// Exam Roll No. - 20220PHY014*

#include <stdio.h>

#include <math.h>

void function(float \**x*, float \**a*, float \**b*)

{

    float y = pow(\**x*, 3) - (4 \* \**x*) - 9;

    if (y < 0)

        \**a* = \**x*;

    else if (y > 0)

        \**b* = \**x*;

}

float absoulte\_difference(float *a*, float *b*)

{

    float absolute = *a* - *b*;

    if (absolute > 0)

        return absolute;

    else

        return -1 \* absolute;

}

int main()

{

    float a = 2, b = 3; *// f(a) < 0 and f(b) > 0*

    float approx;

    int count = 0;

    while (absoulte\_difference(a, b) > 0.0001)

    {

        count++;

        approx = (a + b) / 2;

        function(&approx, &a, &b);

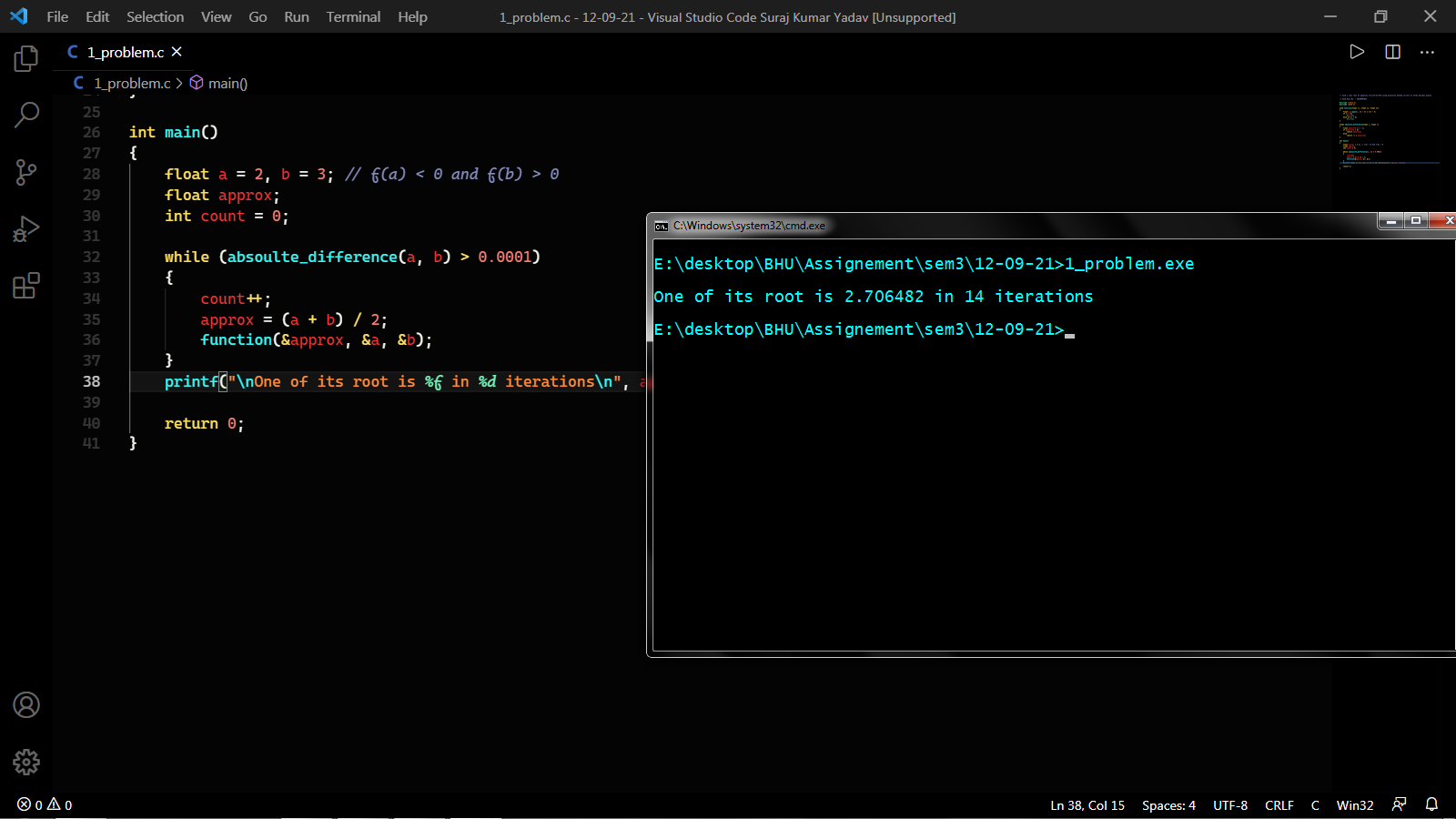
    }

    printf("\nOne of its root is *%f* in *%d* iterations\n", approx, count);

    return 0;

}

Output:



Example 2: *// Find a real root of equation f(x)=x3-x-1=0 using bisection method correct to three decimal places*

*// Exam Roll No. - 20220PHY014*

#include <stdio.h>

#include <math.h>

void function(float \**x*, float \**a*, float \**b*)

{

    float y = pow(\**x*, 3) - (\**x*) - 1;

    if (y < 0)

        \**a* = \**x*;

    else if (y > 0)

        \**b* = \**x*;

}

float absoulte\_difference(float *a*, float *b*)

{

    float absolute = *a* - *b*;

    if (absolute > 0)

        return absolute;

    else

        return -1 \* absolute;

}

int main()

{

    float a = 1, b = 2; *// f(a) < 0 and f(b) > 0*

    float approx;

    int count = 0;

    while (absoulte\_difference(a, b) > 0.0001)

    {

        count++;

        approx = (a + b) / 2;

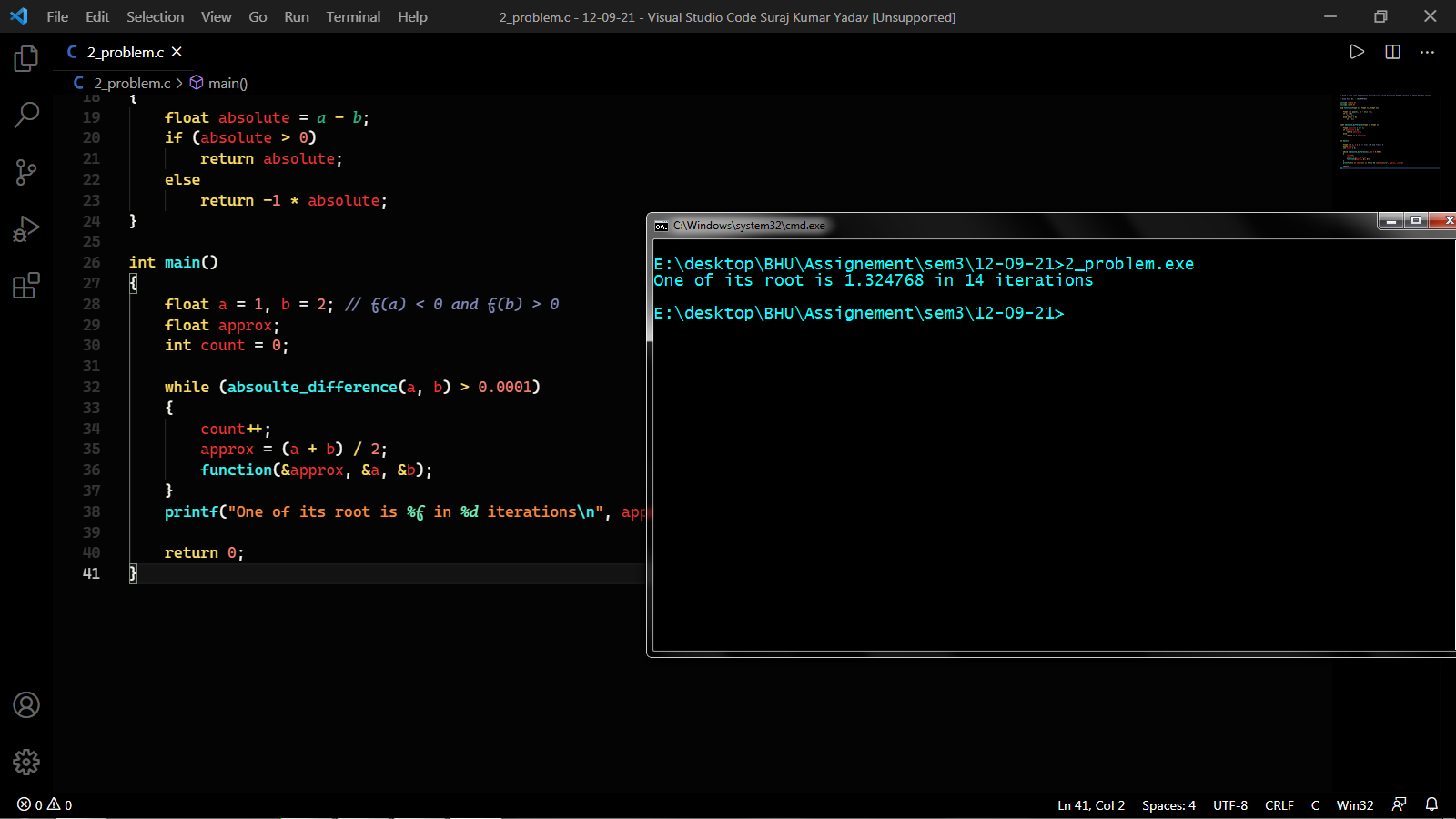
        function(&approx, &a, &b);

    }

    printf("One of its root is *%f* in *%d* iterations\n", approx, count);

    return 0;

}Output:



Practice Problem 1:

*// Find a real root of equation f(x)=x3-2x-5=0 using bisection method correct to two decimal places*

*// Exam Roll No. - 20220PHY014*

#include <stdio.h>

#include <math.h>

void function(float \**x*, float \**a*, float \**b*)

{

    float y = pow(\**x*, 3) - (2 \* \**x*) - 5;

    if (y < 0)

        \**a* = \**x*;

    else if (y > 0)

        \**b* = \**x*;

}

float absoulte\_difference(float *a*, float *b*)

{

    float absolute = *a* - *b*;

    if (absolute > 0)

        return absolute;

    else

        return -1 \* absolute;

}

int main()

{

    float a = 2, b = 3; *// f(a) < 0 and f(b) > 0*

    float approx;

    int count = 0;

    while (absoulte\_difference(a, b) > 0.001)

    {

        count++;

        approx = (a + b) / 2;

        function(&approx, &a, &b);

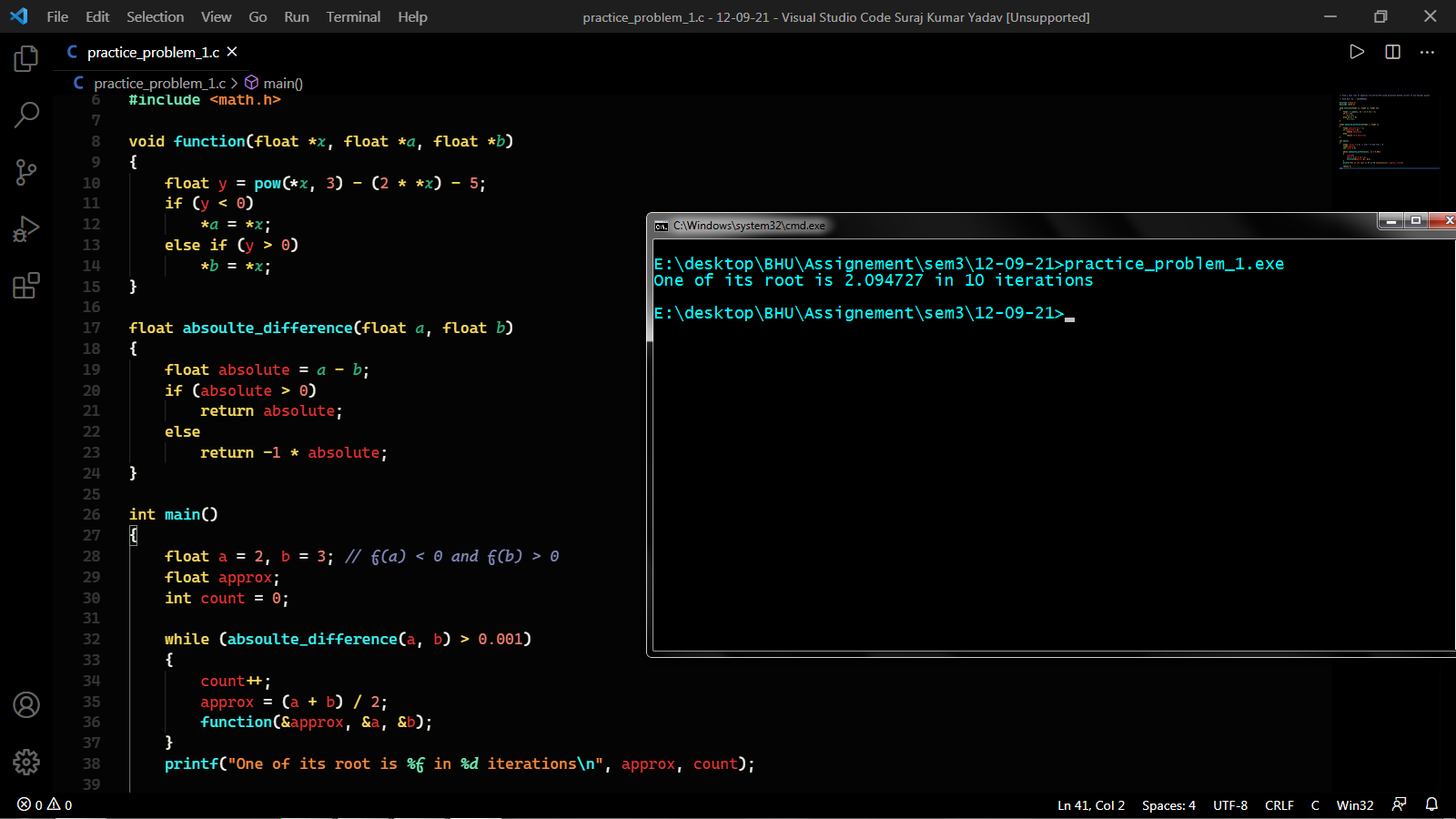
    }

    printf("One of its root is *%f* in *%d* iterations\n", approx, count);

    return 0;

}

Output:



Practice Problem 2:

*// Find a real root of equation f(x)=xe^x-1 using bisection method correct to two decimal places, which lies between 0 and 1*

*// Exam Roll No. - 20220PHY014*

#include <stdio.h>

#include <math.h>

void function(float \**x*, float \**a*, float \**b*)

{

    float y = (\**x* \* pow(M\_E, \**x*)) - 1;

    if (y < 0)

        \**a* = \**x*;

    else if (y > 0)

        \**b* = \**x*;

}

float absoulte\_difference(float *a*, float *b*)

{

    float absolute = *a* - *b*;

    if (absolute > 0)

        return absolute;

    else

        return -1 \* absolute;

}

int main()

{

    float a = 0, b = 1; *// f(a) < 0 and f(b) > 0*

    float approx;

    int count = 0;

    while (absoulte\_difference(a, b) > 0.0001)

    {

        count++;

        approx = (a + b) / 2;

        function(&approx, &a, &b);

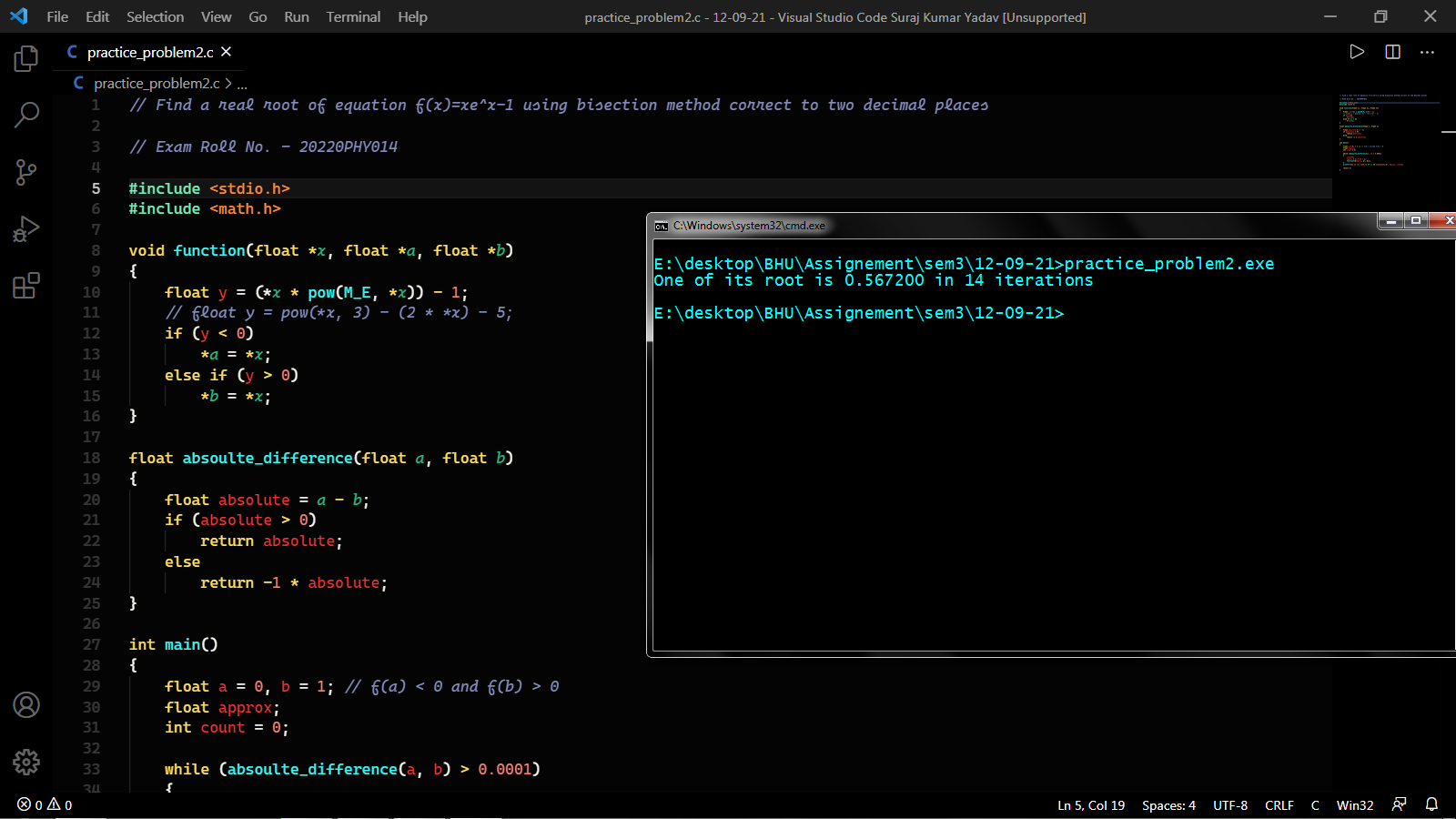
    }

    printf("One of its root is *%f* in *%d* iterations\n", approx, count);

    return 0;

}

Output:



Regula – Falsi Method

Practice Problem 1: