Question 1: *// Write a Program to find the absolute, relative, and percentage error of a given number (2/3), take its three approximations at run time, and tell which one is the best approximation?*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

int main()

{

    float given\_number = (float)2 / 3;

    float approx\_number[3], absolute\_error[3], relative\_error[3], percentage\_error[3];

    for (int i = 0; i < 3; i++)

    {

        printf("Enter approximate number *%d*: ", i + 1);

        scanf("*%f*", &approx\_number[i]);

*// Finding Absolute error*

        absolute\_error[i] = given\_number - approx\_number[i];

        printf("Absolute Error: *%f*\n", absolute\_error[i]);

*// Finding Relative Error:*

        relative\_error[i] = (absolute\_error[i]) / given\_number;

        printf("Relative Error: *%f*\n", relative\_error[i]);

*// Finding Percentage Error:*

        percentage\_error[i] = 100 \* relative\_error[i];

        printf("Percentage Error: *%f*\n", percentage\_error[i]);

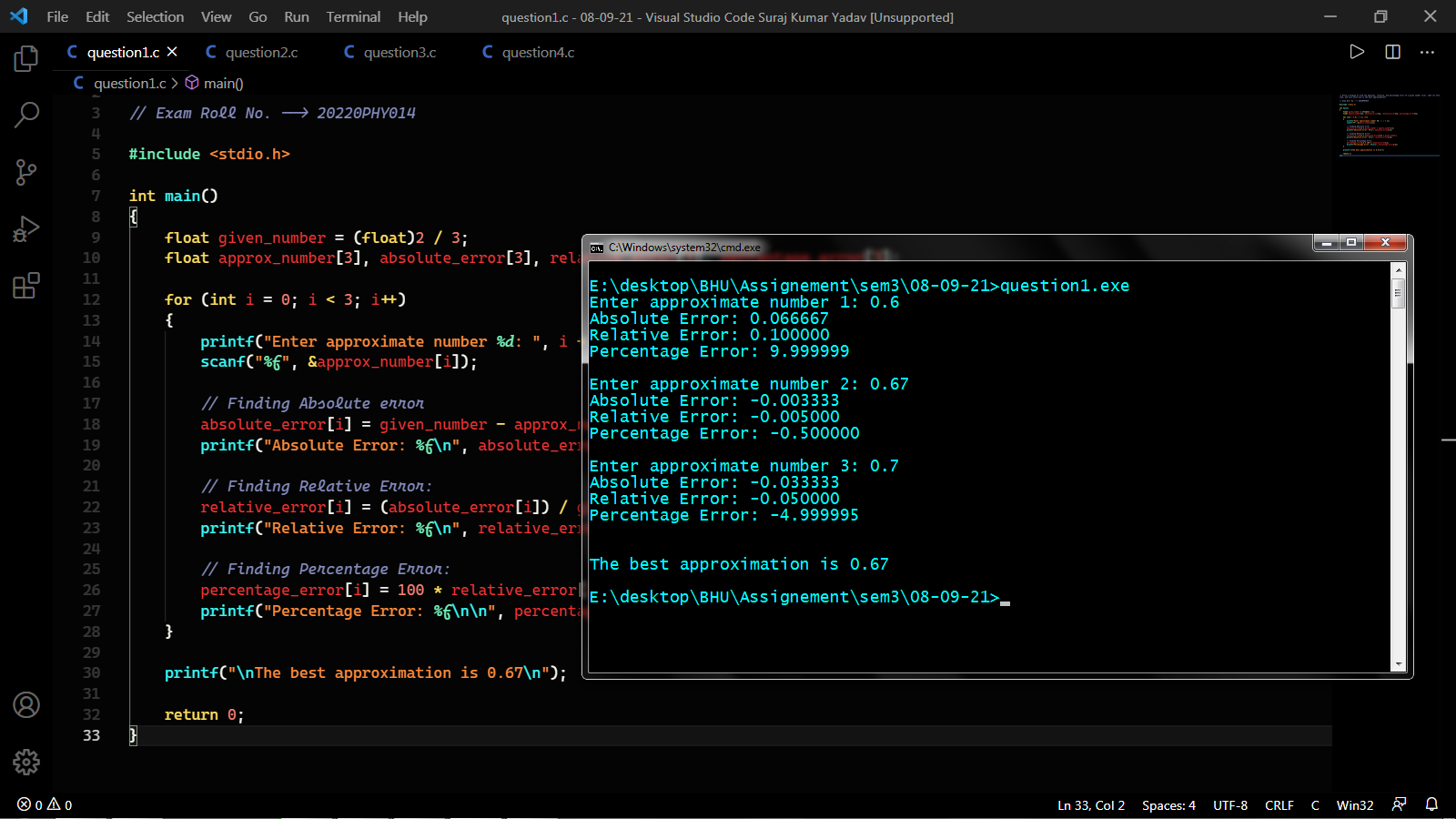
    }

    printf("The best approximation is 0.67\n");

    return 0;

}

Output:



Question 2: *// Write a program to find the root of a Quadratic equation using the Formula method*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

#include <math.h>

int main()

{

    printf("\*\*\* Programme to find all roots of quadratic equation \*\*\*\n\n");

    int a, b, c;

    printf("Enter the values of a, b, c:-\n");

    scanf("*%d%d%d*", &a, &b, &c);

    printf("Your quadratic equation is *%d*x^2 + *%d*x + *%d*\n", a, b, c);

    int d; *// 'd' is used for discriminent*

    d = ((pow(b, 2)) - (4 \* a \* c));

    float r1, r2; *// 'r1', 'r2' means roots*

*// r1 and r2 is for real roots and r3 and r4 is for imaginary roots*

    r1 = (-b + pow(d, 0.5)) / (2 \* a);

    r2 = (-b - pow(d, 0.5)) / (2 \* a);

    if (d > 0)

        printf("This equation have real roots\nAnd its roots are *%f* and *%f*\n", r1, r2);

    else if (d == 0)

        printf("This equation have equal roots\nAnd its root is *%f*\n", r1);

    else

    {

        printf("This equation have imaginary roots\nAnd its roots are (-*%d* + *%.2f*i) / (*%d*)\n", b, sqrt(-d), 2 \* a);

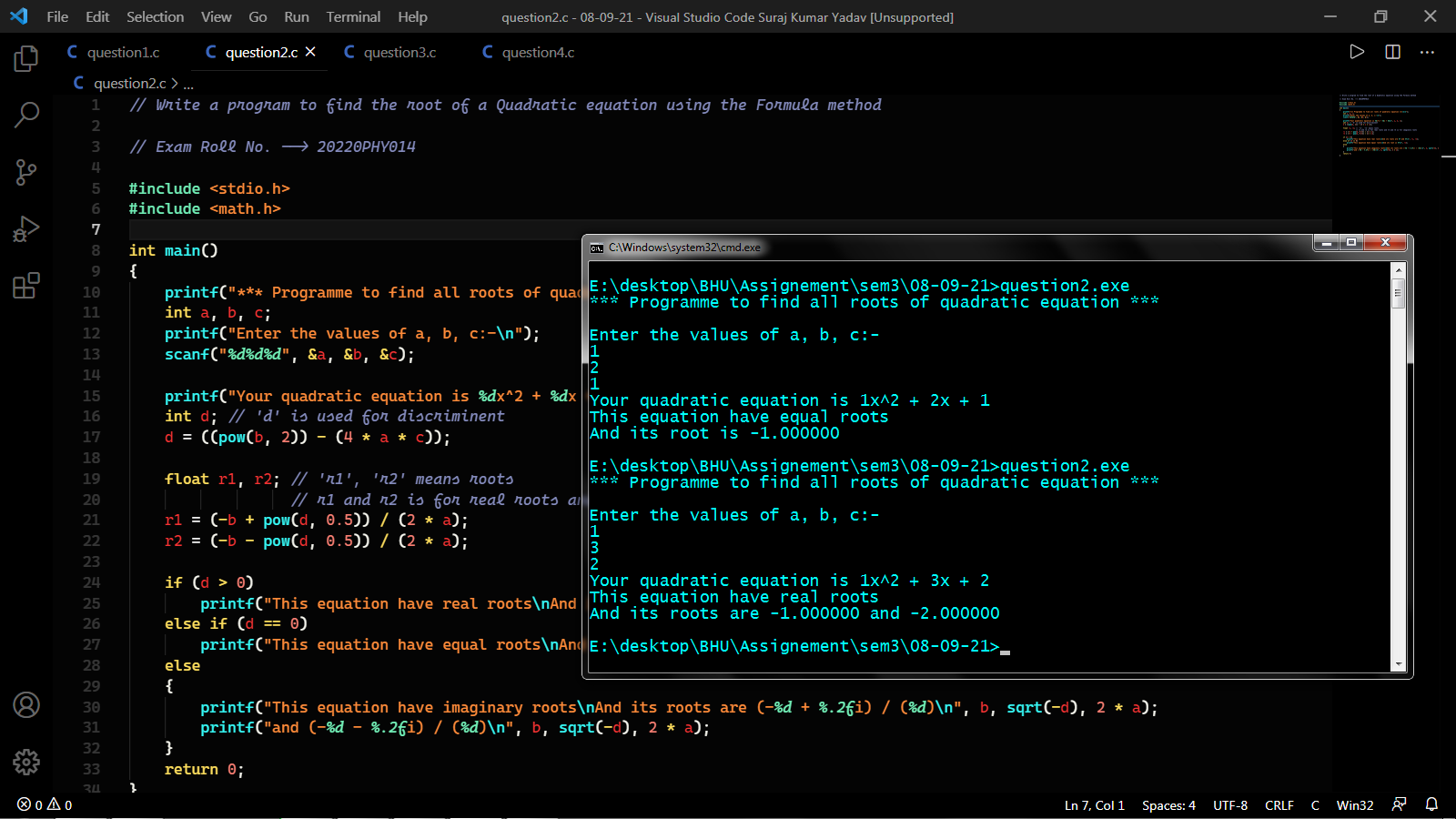
        printf("and (-*%d* - *%.2f*i) / (*%d*)\n", b, sqrt(-d), 2 \* a);

    }

    return 0;

}

Output:



Question 3: */\**

*Write a program to solve the following system of linear equations*

*2x+3y=6*

*4x+9y=15*

*\*/*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

int main()

{

    float x, y;

    int a1, b1, c1, a2, b2, c2;

    puts("Equation 1 is 2x + 3y = 6");

    puts("Equation 2 is 4x + 9y = 15");

*/\**

*For a1x + b1y + c1 = 0 & a2x + b2y + c2 = 0*

*x = (c2\*b1 - c1\*b2) / (a1\*b2 - a2\*b1)*

*y = (c2\*a1 - c1\*a2) / (a2\*b1 - a1\*b2)*

*\*/*

    a1 = 2, b1 = 3, c1 = -6, a2 = 4, b2 = 9, c2 = -15;

    x = (float)(c2 \* b1 - c1 \* b2) / (a1 \* b2 - a2 \* b1);

    y = (float)(c2 \* a1 - c1 \* a2) / (a2 \* b1 - a1 \* b2);

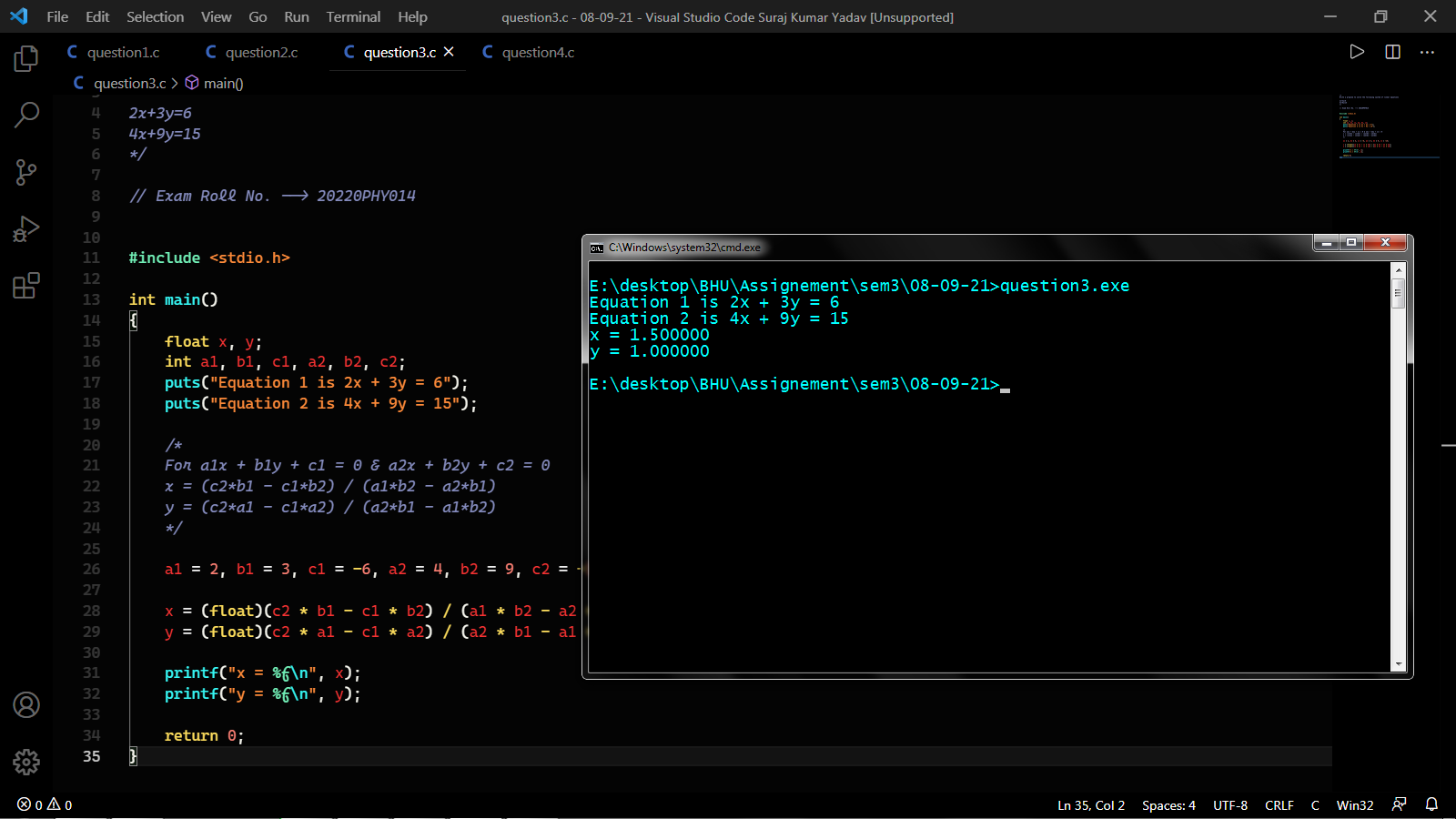
    printf("x = *%f*\n", x);

    printf("y = *%f*\n", y);

    return 0;

}

Output:



Question 4: *// Write a program to find the sum of the first 10 terms of a "sine series"*

*// sin(x) = x - x^3/3! + x^5/5! - ...*

*// Exam Roll No. --> 20220PHY014*

#include <stdio.h>

#include <math.h>

int factorial(int *num*);

int main()

{

    int power = 1;    *// power of x in series*

    float x, sum = 0; *// x is angle of sine and sum if the sum upto 10th term of the series*

    printf("Enter x in degree: ");

    scanf("*%f*", &x);

*// converting x to radian*

    x = (x \* M\_PI) / 180;

    printf("sin(*%f*) = ", x);

    for (int i = 0; i < 10; i++)

    {

        sum += (pow(-1, i) \* (pow(x, power) / factorial(power)));

        power += 2;

    }

    printf("*%f*\n", sum);

    return 0;

}

int factorial(int *num*)

{

    int fact = 1;

    for (int i = *num*; i > 1; i--)

        fact \*= i;

    return fact;

}

Output:

