

ASSIGNMENT-3(LIFT OFF-C)

GROUP-11

NAME-BISWO RANJAN PRADHAN

BISWO RANJAN PRADHAN

REG NO.-2002020031

BRANCH-CHEMICAL ENGINEERING

The screenshot shows the Visual Studio Code editor with the file explorer on the left. The 'OPEN EDITORS' list includes 'Welcome', 'a.exe', 'quone.c', 'qutwo.c', 'quthree.c', and 'qufour.c'. The 'ASSIGNMENT-3' folder contains 'a.exe', 'qufour.c', 'quone.c', 'quthree.c', and 'qutwo.c'. The main editor window displays the code for 'quone.c'.

```
1 //write a C program to find the square of any number using the function
2
3 #include<stdio.h>
4
5 int func(int);
6
7 int main()
8 {
9     int no, square;
10
11     printf("\n Enter an no : ");
12     scanf("%d",&no);
13
14     square = func(no);
15
16     printf("\n Square of no is : %d ", square);
17 }
18
19
20 int func(int temp)
21 {
22     return temp*temp;
23 }
24 |
```

The status bar at the bottom indicates 'Ln 24, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Win32', and the date '23:18 02-03-2021'.

BISWO RANJAN PRADHAN

The screenshot shows the Visual Studio Code terminal window. The terminal output is as follows:

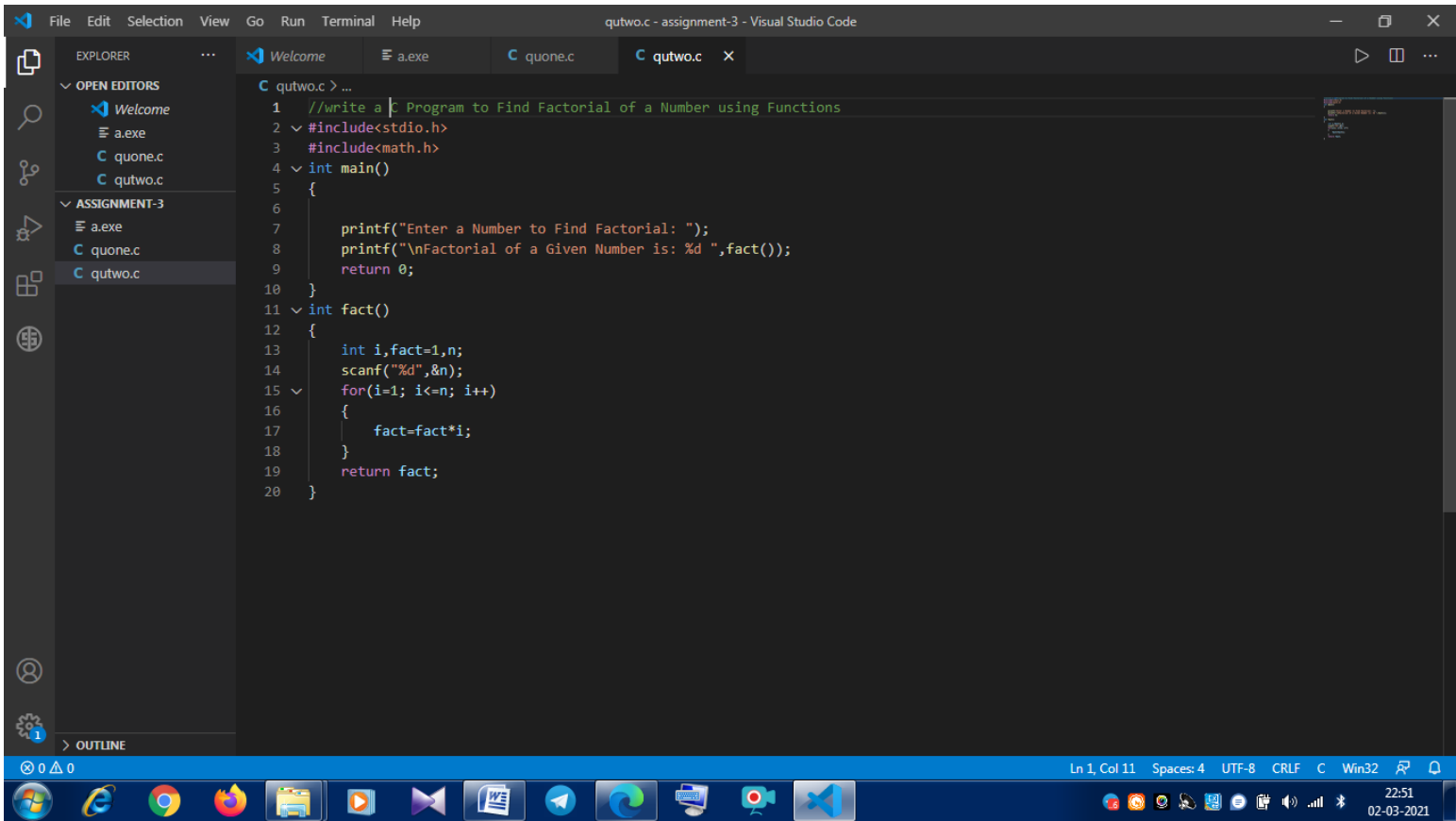
```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\DEll\assignment-3> gcc quone.c
PS C:\Users\DEll\assignment-3> .\a.exe

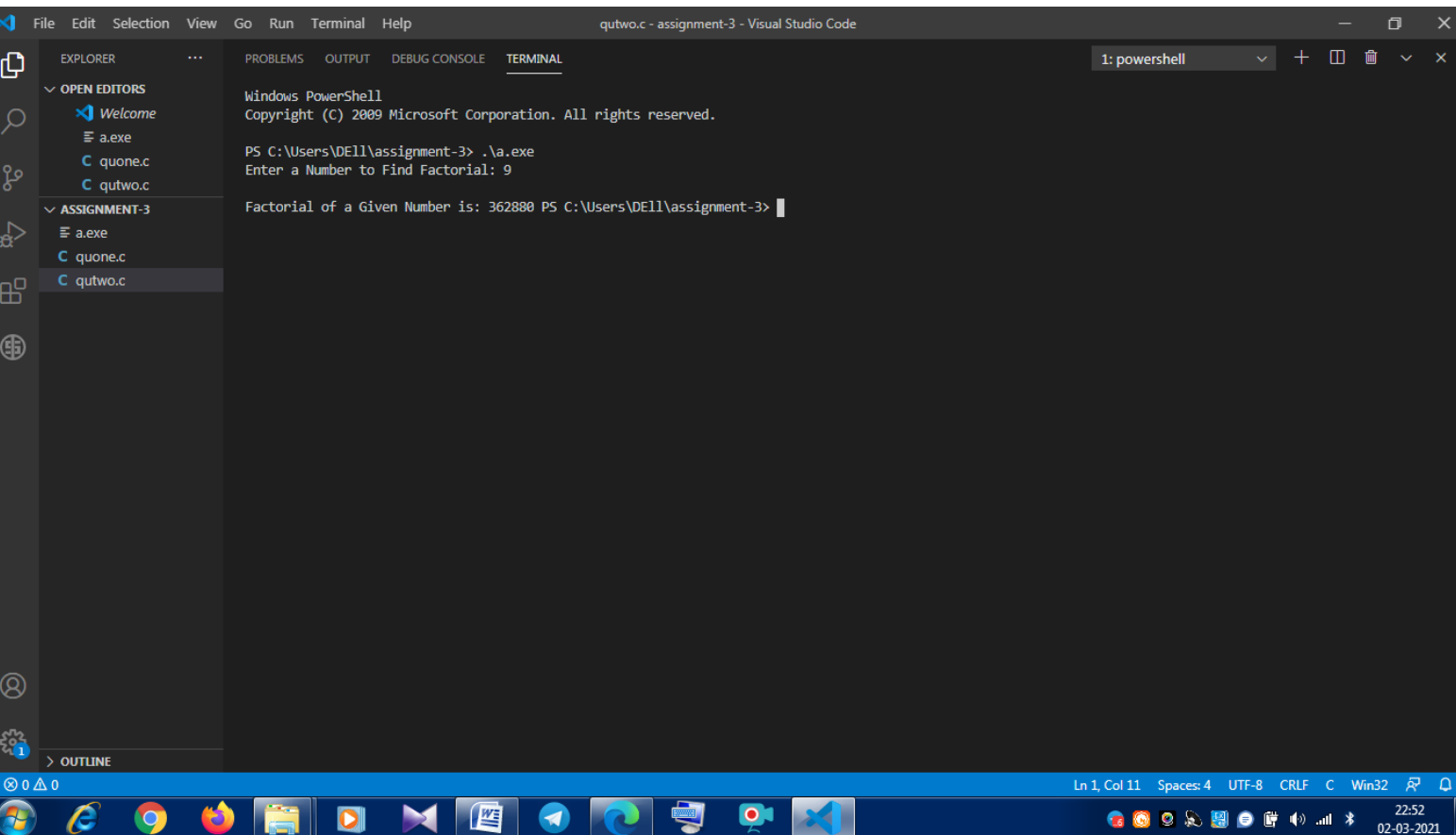
Enter an no : 28

Square of no is : 784 PS C:\Users\DEll\assignment-3> |
```

The status bar at the bottom indicates 'Ln 24, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'C', 'Win32', and the date '23:19 02-03-2021'.



BISWO RANJAN PRADHAN



The screenshot shows the Visual Studio Code editor with a C program named `quthree.c` open. The program is designed to find the Greatest Common Divisor (GCD) and Least Common Multiple (LCM) of two integers using recursion. The code includes a function prototype for `gcd`, a `main` function that takes user input, and a recursive function `gcd` that calculates the GCD. The LCM is then calculated as $(\text{num1} * \text{num2}) / \text{hcf}$.

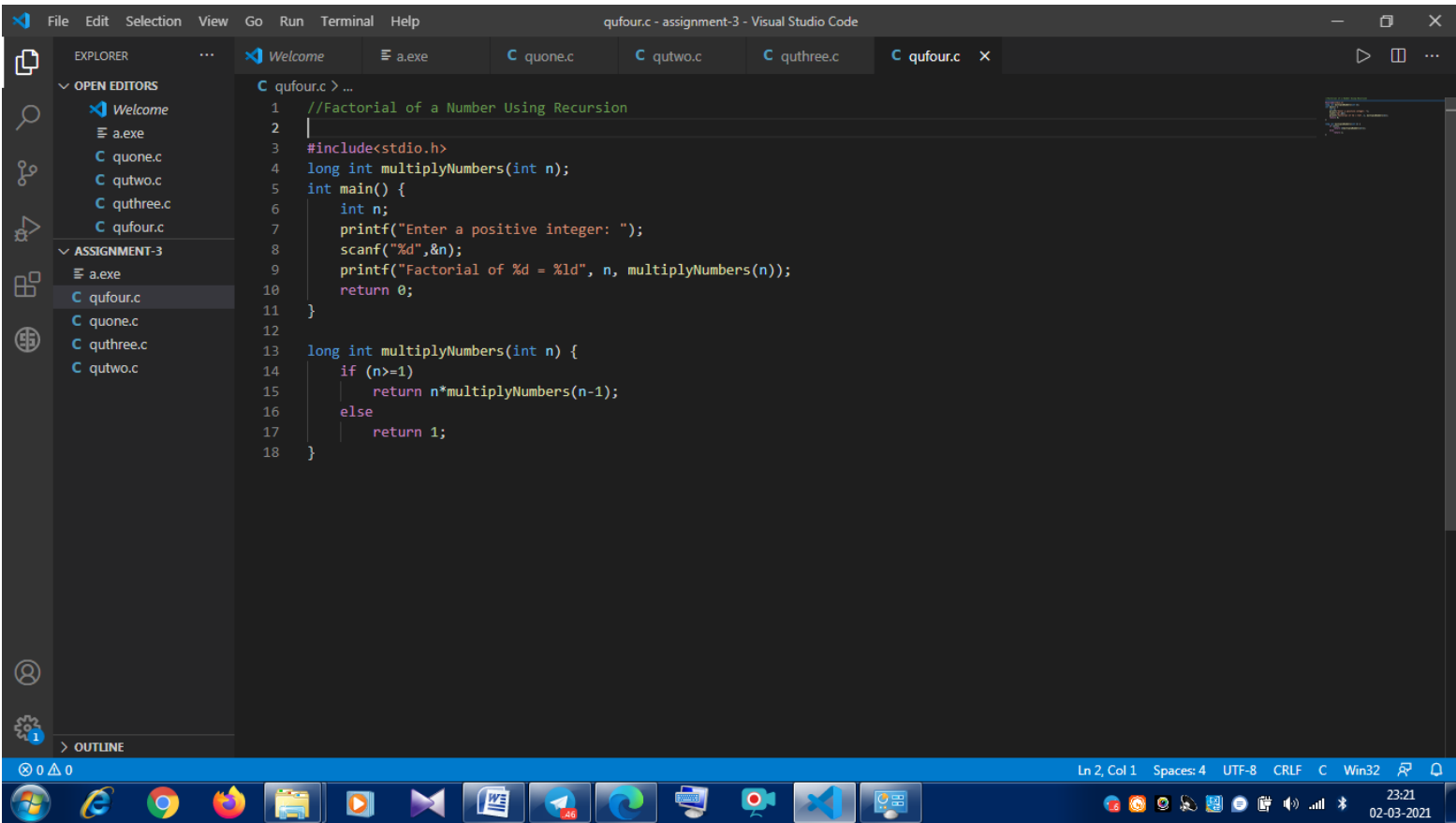
```
1 //write a C program to find GCD and LCM using recursion
2 #include <stdio.h>
3 int gcd(int x, int y); //function prototype
4
5 int main()
6 {
7     int num1, num2, hcf, lcm;
8
9     printf("Enter two integer Values:\n");
10    scanf("%d %d", &num1, &num2);
11
12    hcf = gcd(num1, num2);
13    printf("GCD: %d", hcf);
14    printf("\nLCM: %d", (num1 * num2) / hcf);
15    return 0;
16 }
17 //recursive function
18 int gcd(int x, int y)
19 {
20     if (y == 0) //recursion termination condition
21     {
22         return x;
23     }
24     else
25     {
26         return gcd(y, x % y); //calls itself
27     }
28 }
```

BISWO RANJAN PRADHAN

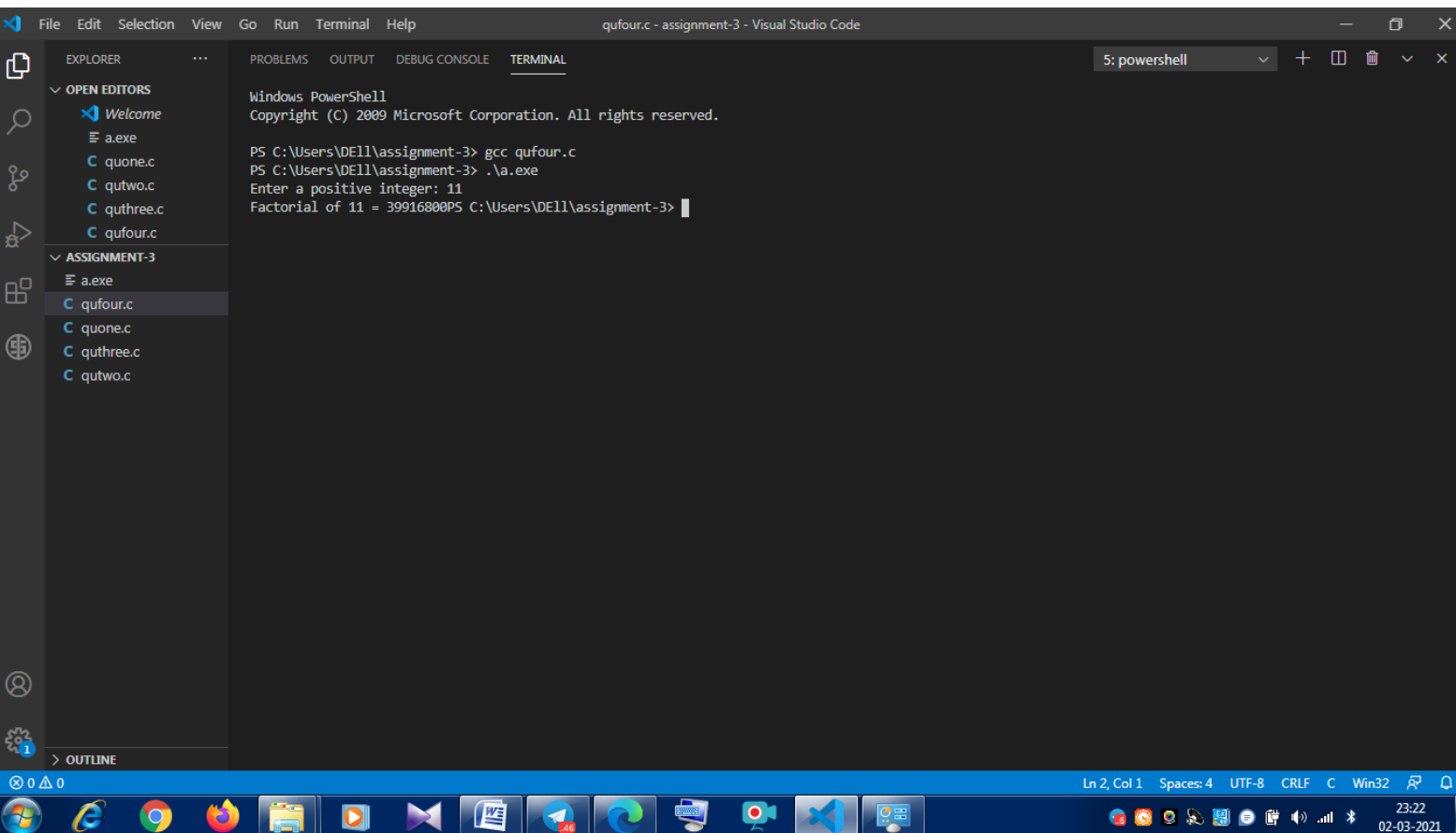
The screenshot shows the Visual Studio Code terminal window with the command prompt. The user has compiled the `quthree.c` file using `gcc` and executed it using `./a.exe`. The program prompts the user to enter two integer values, 15 and 12. The output shows the GCD as 3 and the LCM as 60.

```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\DELL\assignment-3> gcc quthree.c
PS C:\Users\DELL\assignment-3> .\a.exe
Enter two integer Values:
15
12
GCD: 3
LCM: 60PS C:\Users\DELL\assignment-3>
```



BISWO RANJAN PRADHAN



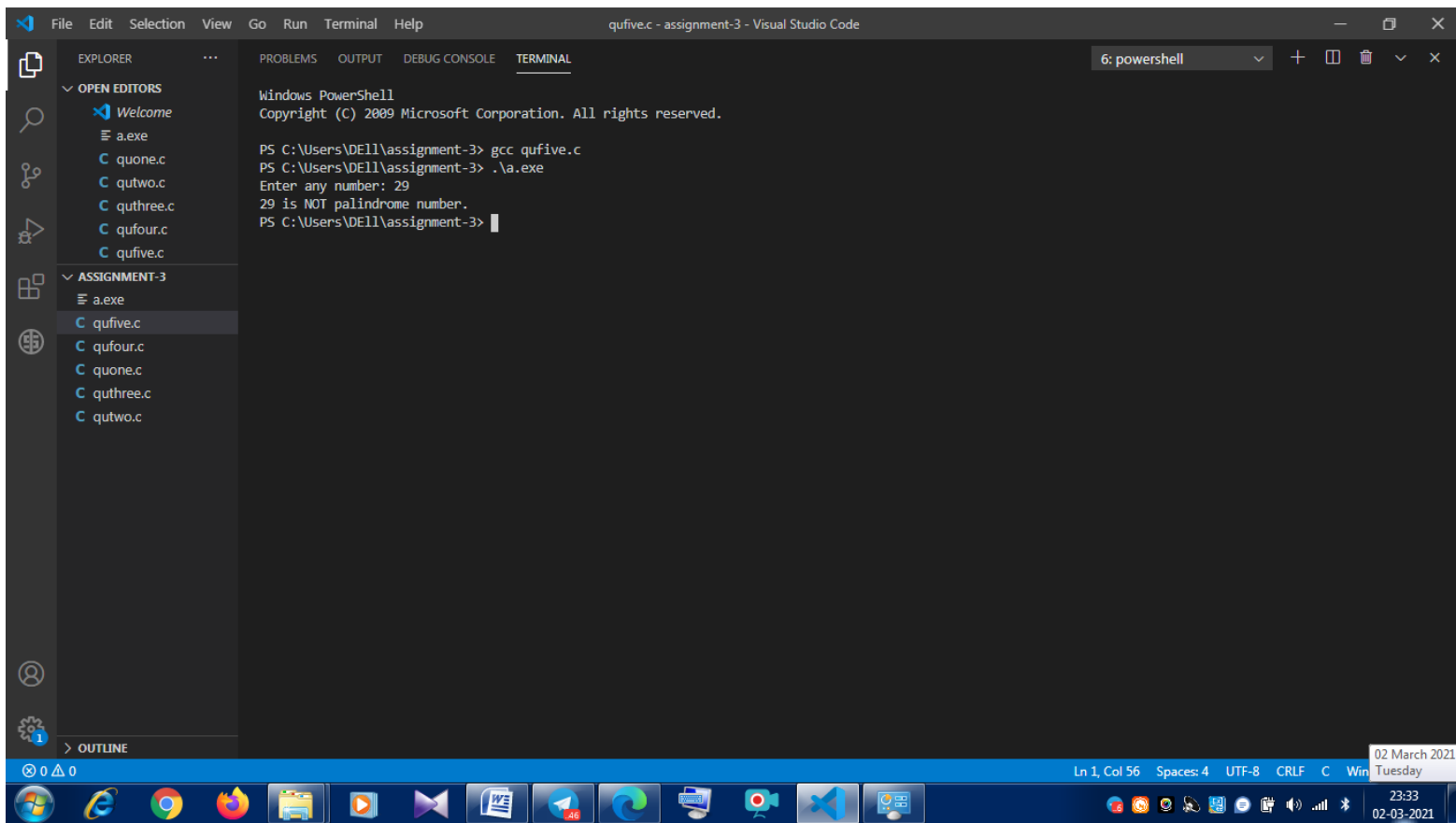
This screenshot shows the Visual Studio Code editor with a C program named `qufive.c` open. The program is designed to check if a number is a palindrome using recursion. The code includes standard headers, function declarations for `reverse` and `isPalindrome`, and a `main` function that prompts the user for input and prints the result.

```
1 // C program to check palindrome number using recursion
2
3
4 #include <stdio.h>
5 #include <math.h>
6
7
8 /* Function declarations */
9 int reverse(int num);
10 int isPalindrome(int num);
11
12
13
14 int main()
15 {
16     int num;
17
18     /* Input any number from user */
19     printf("Enter any number: ");
20     scanf("%d", &num);
21
22     if(isPalindrome(num) == 1)
23     {
24         printf("%d is palindrome number.\n", num);
25     }
26     else
27     {
28         printf("%d is NOT palindrome number.\n", num);
29     }
30
31     return 0;
32 }
```

BISWO RANJAN PRADHAN

This screenshot shows the implementation of the `reverse` and `isPalindrome` functions in the `qufive.c` file. The `isPalindrome` function calls `reverse` to compare the original number with its reverse. The `reverse` function uses a recursive approach to calculate the reverse of the number by finding the number of digits and then building the reversed number.

```
35
36 /**
37  * Function to check whether a number is palindrome or not.
38  * This function returns 1 if the number is palindrome otherwise 0.
39  */
40 int isPalindrome(int num)
41 {
42     /*
43      * Check if the given number is equal to
44      * its reverse.
45      */
46     if(num == reverse(num))
47     {
48         return 1;
49     }
50     return 0;
51 }
52
53
54
55 /**
56  * Recursive function to find reverse of any number
57  */
58 int reverse(int num)
59 {
60     /* Find number of digits in num */
61     int digit = (int)log10(num);
62
63     /* Recursion base condition */
64     if(num == 0)
65         return 0;
66
67     return ((num%10 * pow(10, digit)) + reverse(num/10));
68 }
```



BISWO RANJAN PRADHAN

THANK YOU