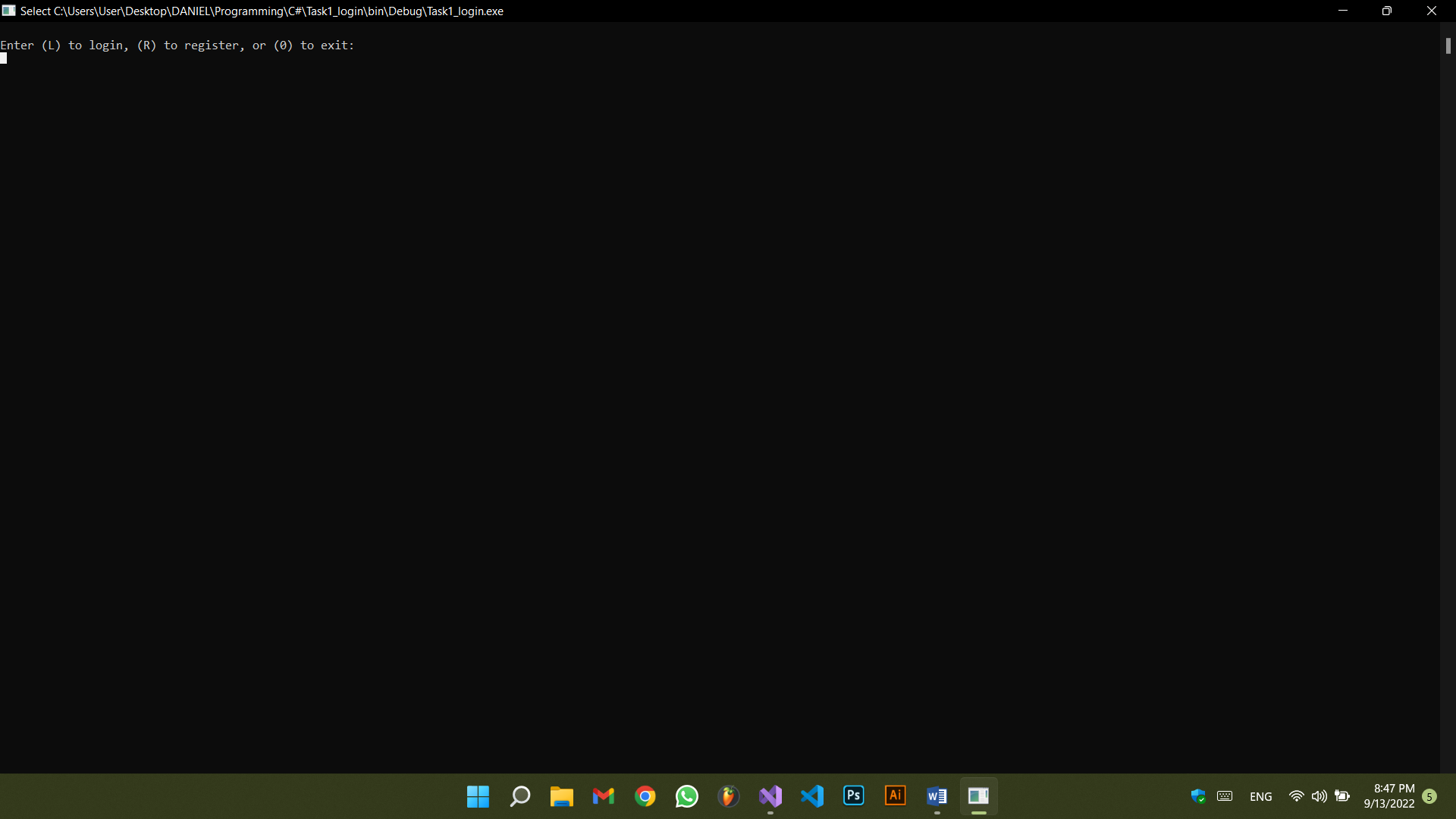
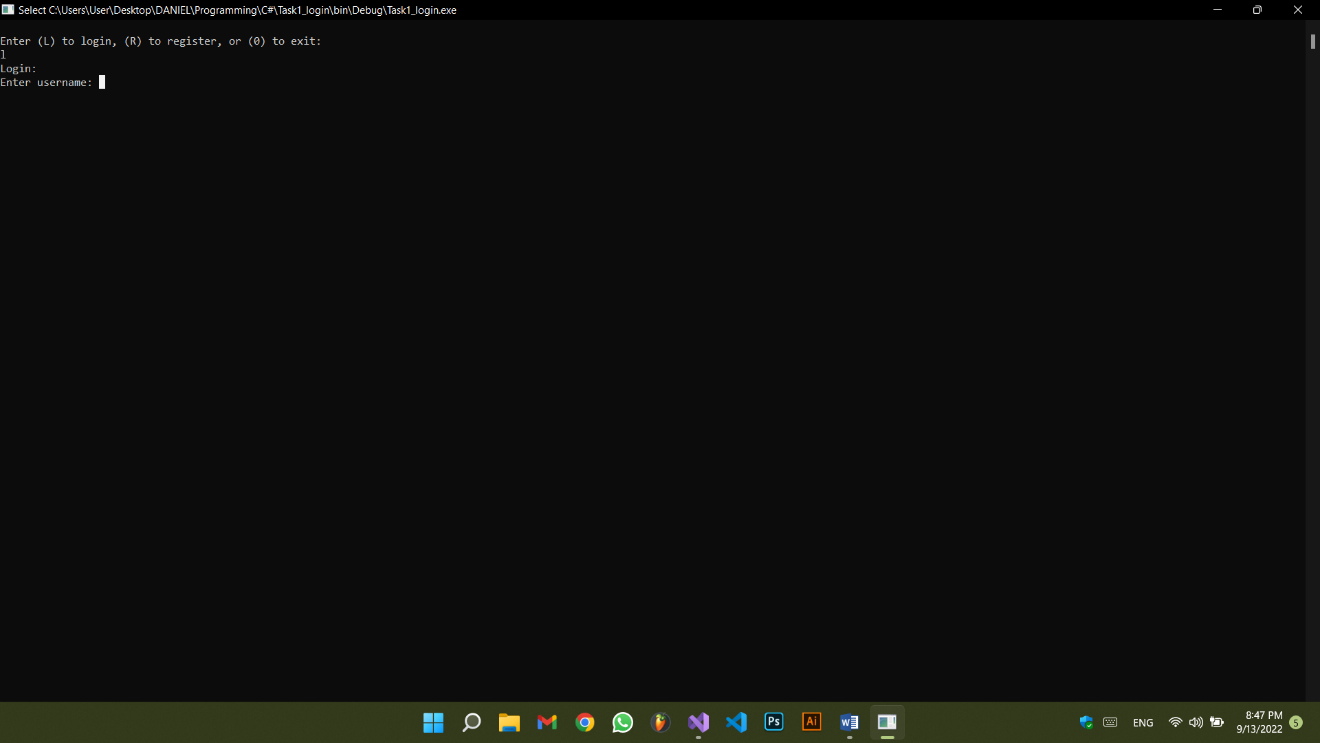
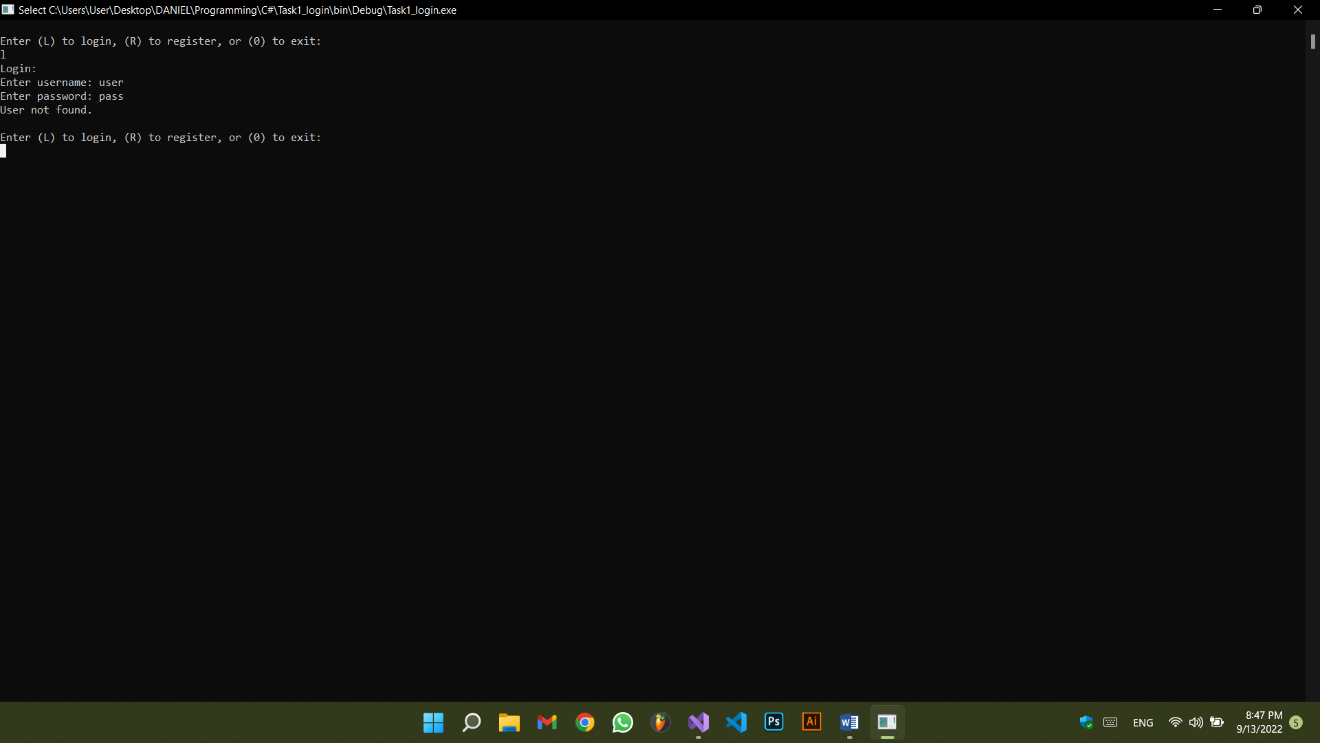
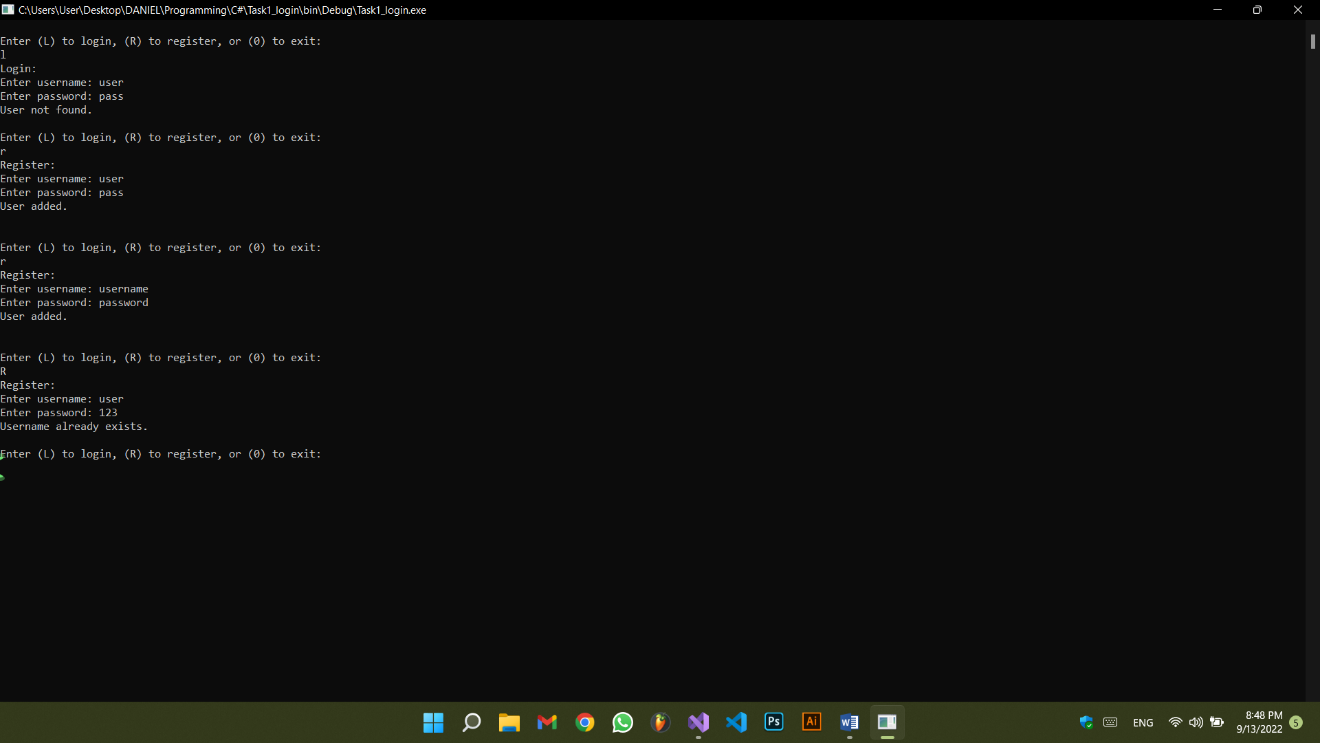
***Part 1:***

**Task 1:**

I used the switch() statement to check the choice of the user, L to login, R to register and 0 to exit. The L case calls the login function that takes a username (key) and a password (value) from the user and checks all the elements of the dictionary using a foreach loop. If the username and password are found, a welcome message is displayed. If not, an error message is displayed.

The R case calls the register function that takes a username (key) and a password (value) from the user and adds the pair to the dictionary. If the user enters an already existing username, an error message is displayed.

The 0 case terminates the program.

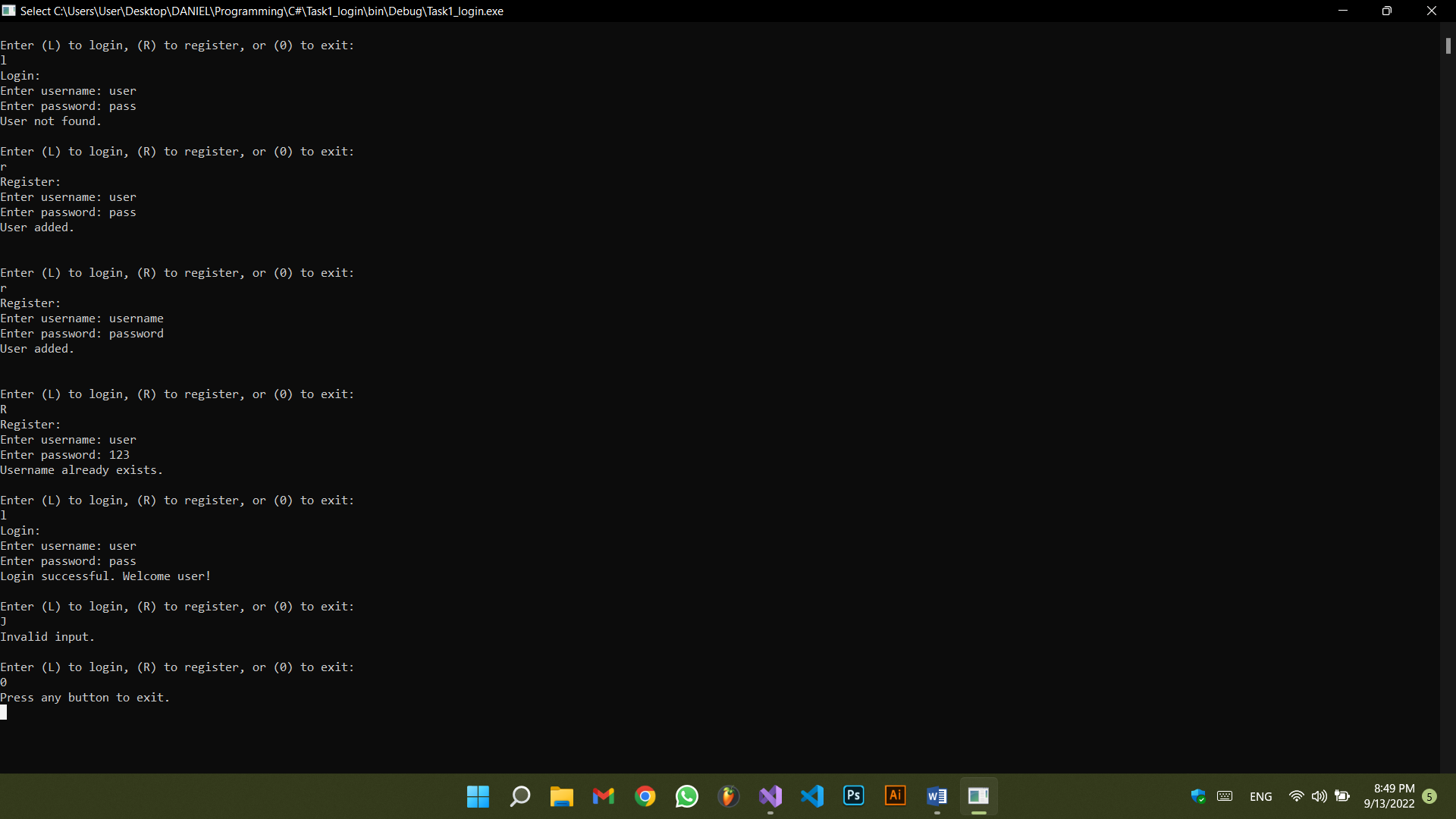
If the user inputs any other character, an error message is displayed and the user is asked again (while loop) to choose to either login, register, or exit.

4

3

2

1



5

using System;

using System.Collections.Generic;

namespace Task1\_login

{

internal class Program

{

static void Main(string[] args)

{

Dictionary<string, string> dict = new Dictionary<string, string>();

char choice; //login, register or exit.

while(true) //the program will close only if the user chooses to exit.

{

Console.WriteLine("\nEnter (L) to login, (R) to register, or (0) to exit: ");

choice = char.Parse(Console.ReadLine());

switch (choice)

{

case 'L':

case 'l':

login(ref dict);

break;

case 'R':

case 'r':

registerPair(ref dict);

break;

case '0':

Console.WriteLine("Press any button to exit.");

Console.ReadKey();

return;

default:

Console.WriteLine("Invalid input.");

break;

}

}

}

static void registerPair(ref Dictionary<string, string> dict)

{

Console.WriteLine("Register:");

Console.Write("Enter username: ");

string username = Console.ReadLine();

Console.Write("Enter password: ");

string password = Console.ReadLine();

bool search = false;

foreach (KeyValuePair<string, string> pair in dict) //checking is username already exists.

{

if (pair.Key == username)

{

search = true;

break;

}

}

if (search == false)

{

dict.Add(username, password);

Console.WriteLine("User added.\n");

}

else Console.WriteLine("Username already exists.");

return;

}

static void login(ref Dictionary<string, string> dict)

{

Console.WriteLine("Login:");

Console.Write("Enter username: ");

string username = Console.ReadLine();

Console.Write("Enter password: ");

string password = Console.ReadLine();

foreach(KeyValuePair<string, string> pair in dict) //checking every pair of the dictionary.

{

if (pair.Key == username && pair.Value == password)

{

Console.WriteLine("Login successful. Welcome {0}!", pair.Key);

return;

}

}

Console.WriteLine("User not found."); //is displayed if the dictionary is empty and if input is invalid.

return;

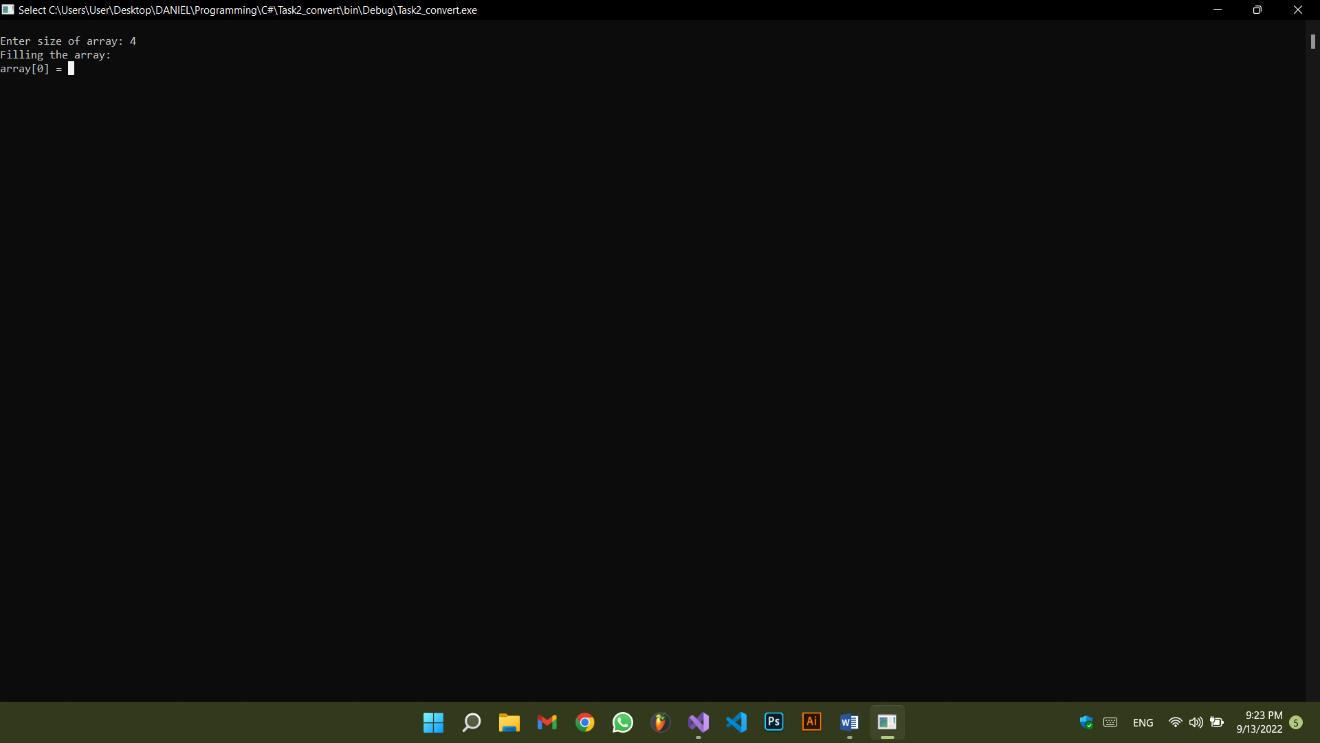
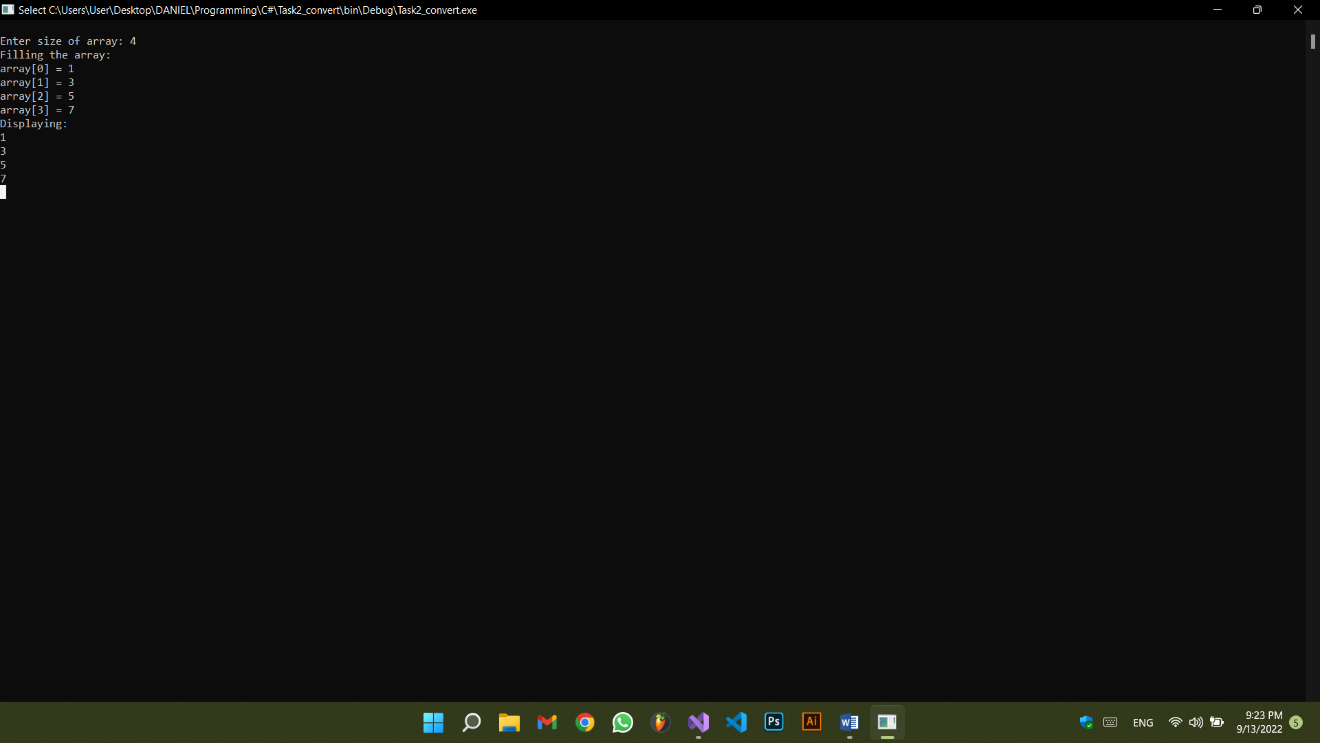
}

}

}

**Task 2:**

The fillArray function was called so that the user fills an array. Then the convertToList is called to access every element of the array and add it to a list. That list is returned by the convertToArray method and then the list is displayed.



using System;

using System.Collections.Generic;

namespace Task2\_convert

{

class Program

{

static void Main(string[] args)

{

Console.Write("\nEnter size of array: ");

int size = int.Parse(Console.ReadLine());

int[] arr = new int[size];

fillArray(ref arr);

List<int> list = convertToList(arr);

Console.WriteLine("Displaying:");

for (int i = 0; i < arr.Length; i++)

Console.WriteLine(list[i]);

Console.ReadKey();

}

static List<int> convertToList(int[] array)

{

List<int> list = new List<int>();

for (int i = 0; i < array.Length; i++) //accessing every element in array.

{

list.Add(array[i]); //adding each element to the list.

}

return list;

}

static void fillArray(ref int[] array)

{

Console.WriteLine("Filling the array:");

for(int i =0; i < array.Length; i++)

{

Console.Write("array[{0}] = ", i);

array[i] = int.Parse(Console.ReadLine());

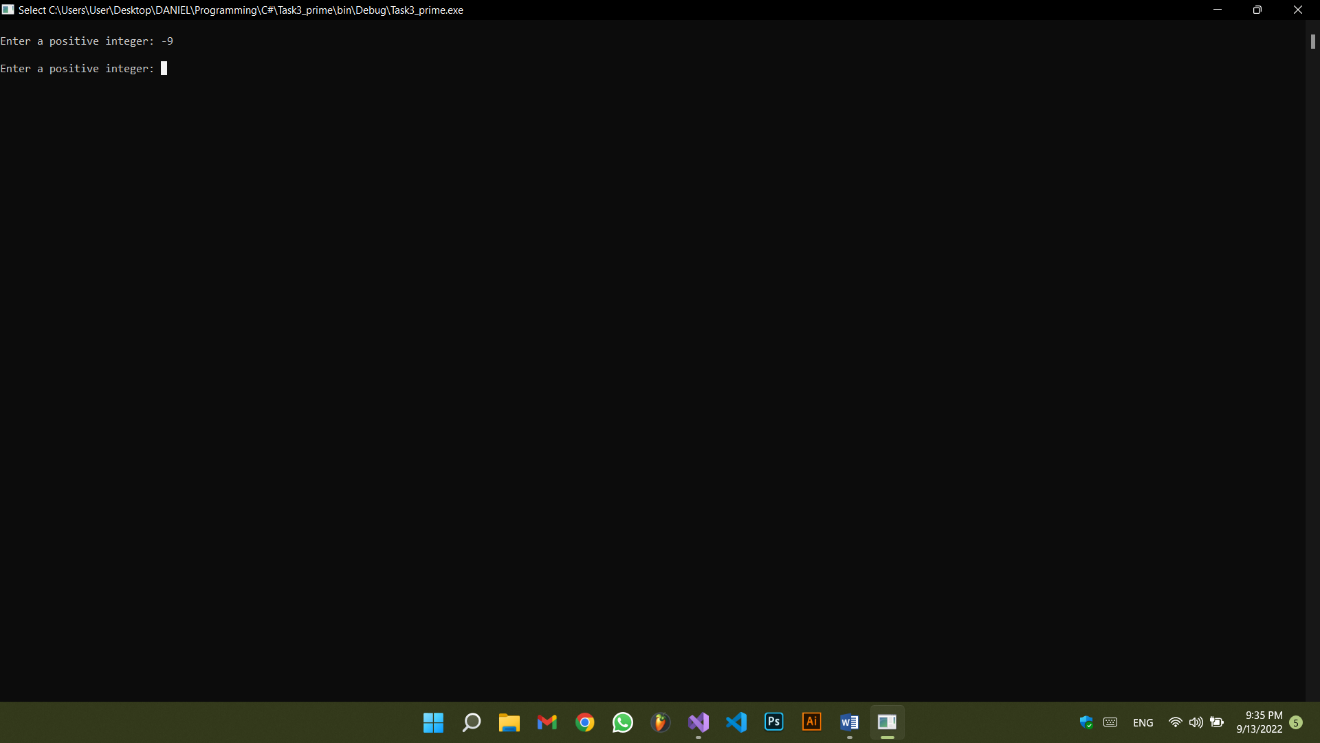
}

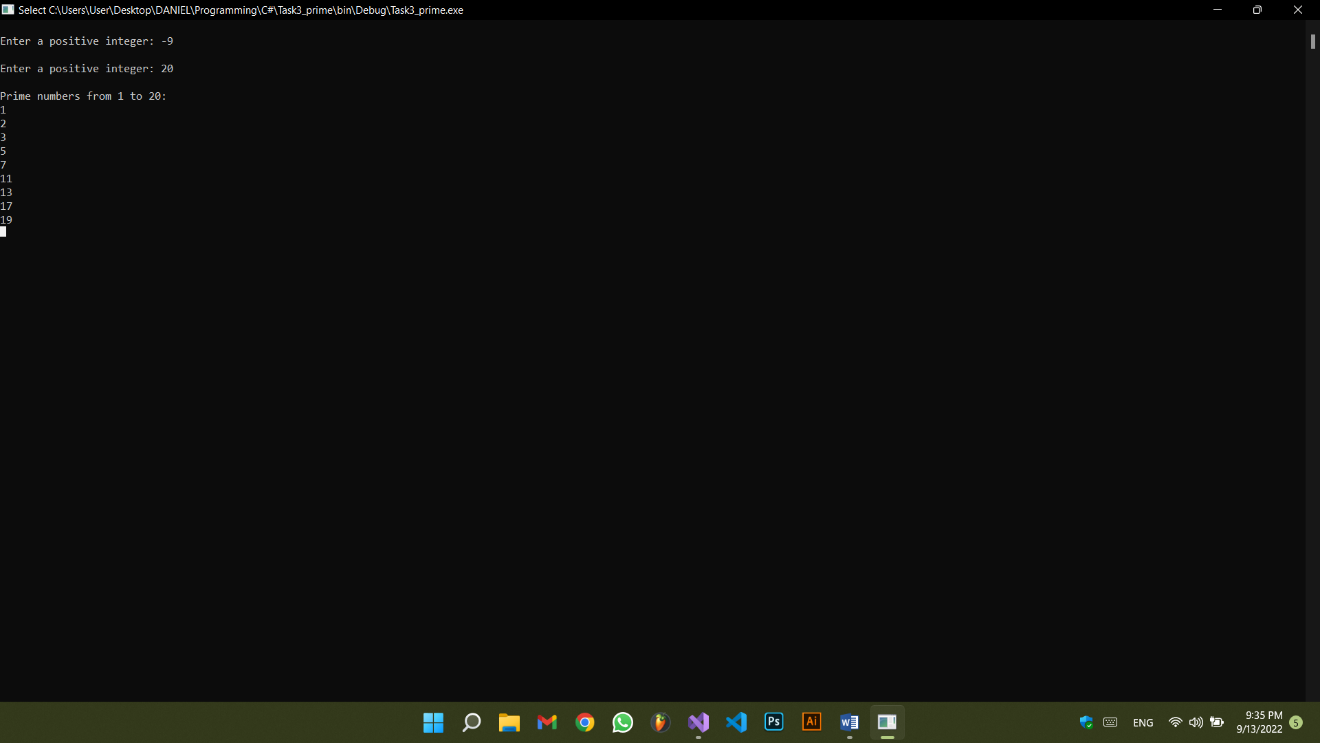
}

}

}

**Task 3:**

A positive number (num) is entered by the user (do while loop), and then the displayPrime method is called, it checks every number from 1 to num (for loop), then each number (j) is checked if it has divisors other than one and itself using a for loop that traverses from 2 to i-1 to check if that j is a divisor of I or not. If no divisors are found for j, then this j is displayed.



using System;

namespace Task3\_prime

{

class Program

{

static void displayPrime(int x)

{

bool hasDivisors; //used to determine if a number has divisors other than 1 and itself.

Console.WriteLine("\nPrime numbers from 1 to {0}:", x);

for (int i = 1; i <= x; i++)

{

hasDivisors = false;

for (int j = 2; j < i; j++) //every number is divisible by 1 and itself, j checks every number from 2 to i-1.

{

if (i % j == 0)

{

hasDivisors = true;

break;

}

}

if (!hasDivisors)

Console.WriteLine(i); //i is only printed if no j was a divisor of i.

}

}

static void Main(string[] args)

{

int num;

do

{

Console.Write("\nEnter a positive number: ");

num = int.Parse(Console.ReadLine()); //converts the string returned from ReadLine into an integer.

} while (num < 0);

//displaying:

displayPrime(num);

Console.ReadKey();

}

}

}