DAY-1

# *1.To check the given number is even or odd:*

#include <stdio.h>

int main() {

int num;

printf("Enter an integer: ");

scanf("%d", &num);

if(num % 2 == 0)

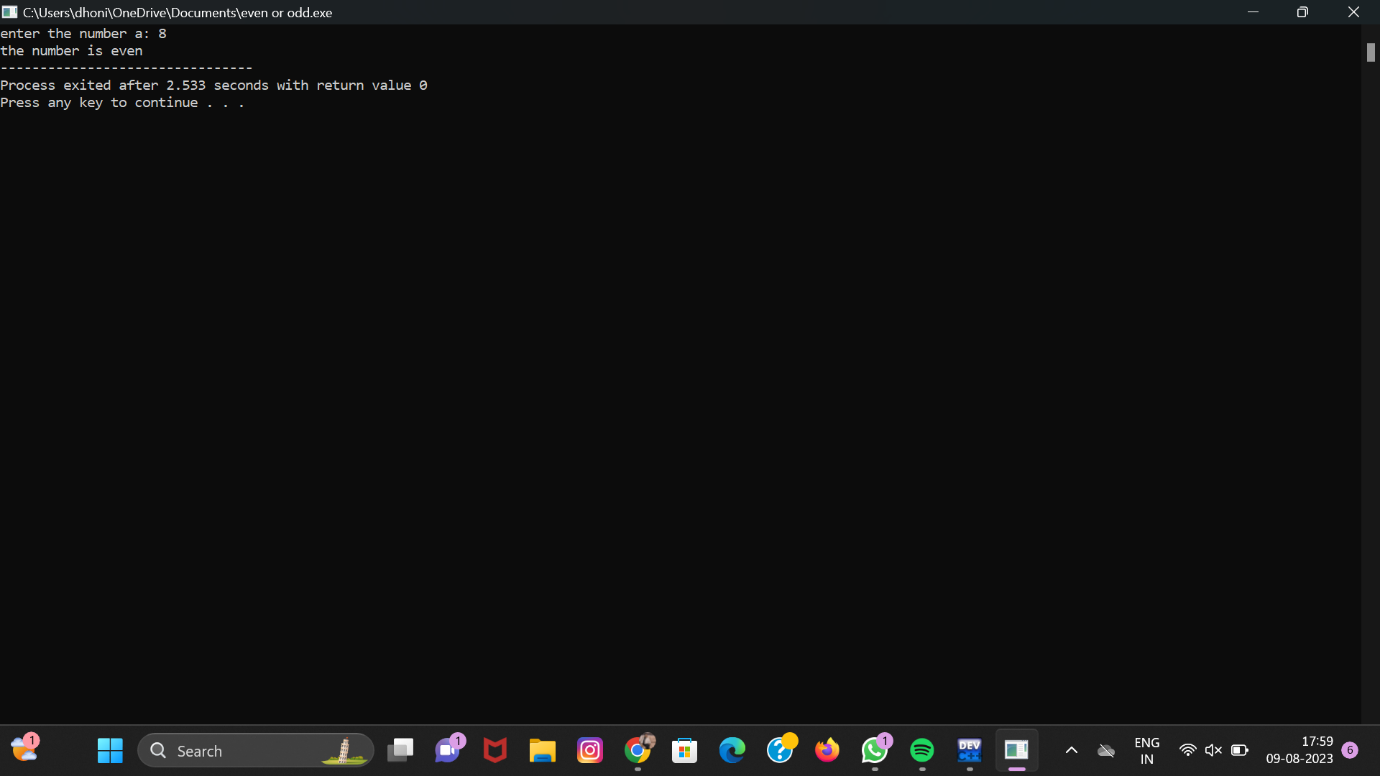
printf("%d is even.", num);

else

printf("%d is odd.", num);

return 0;

}



# *2.Find the sum of first n numbers using for loop:*

#include <stdio.h>

#include <conio.h>

int main()

{

int num, i, sum = 0;

printf(" Enter a positive number: ");

scanf("%d", &num);.

for (i = 0; i <= num; i++)

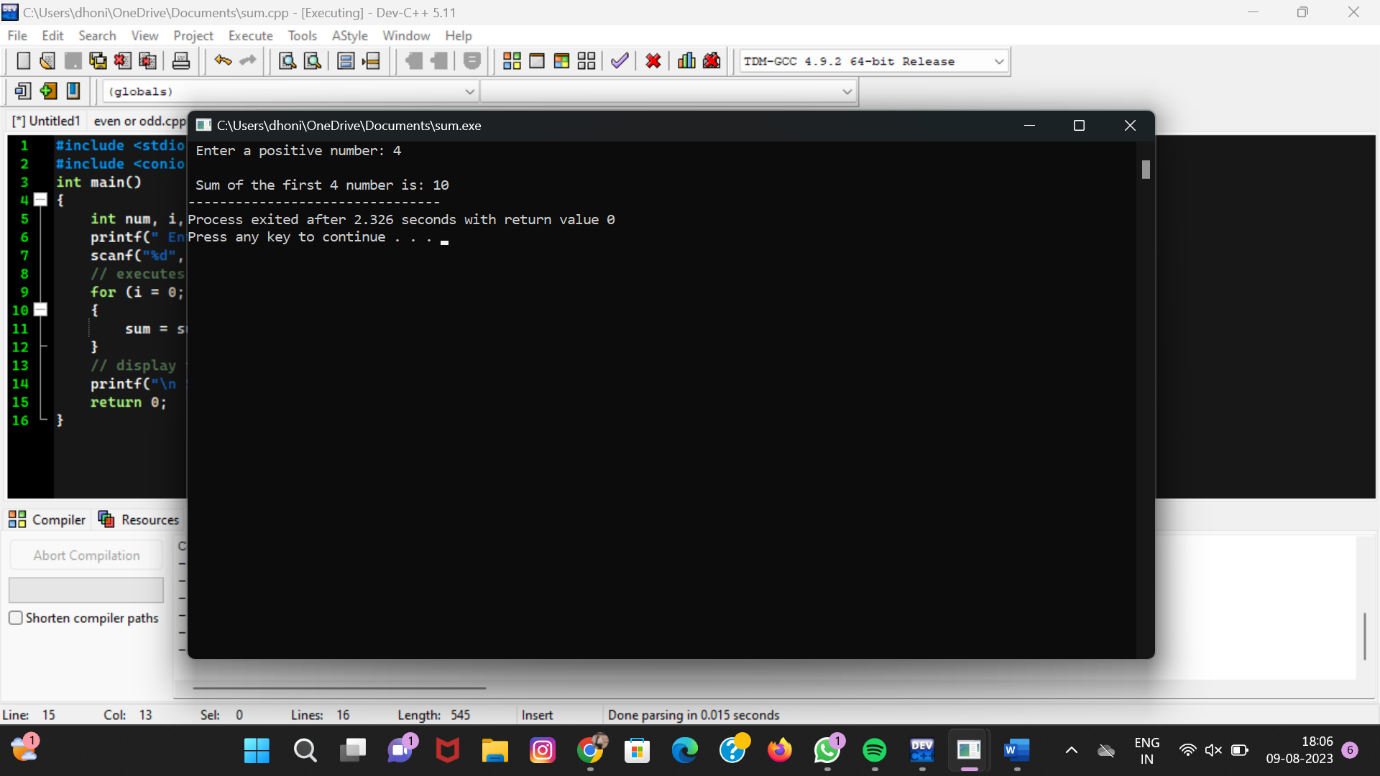
{

sum = sum + i;

}

printf("\n Sum of the first %d number is: %d", num, sum);

return 0;

} 

# *3.To find sum of even numbers :*

#include<stdio.h>

int main()

{

int i, n, sum=0;

printf("Enter upper limit: ");

scanf("%d", &n);

for(i=2; i<=n; i+=2)

{

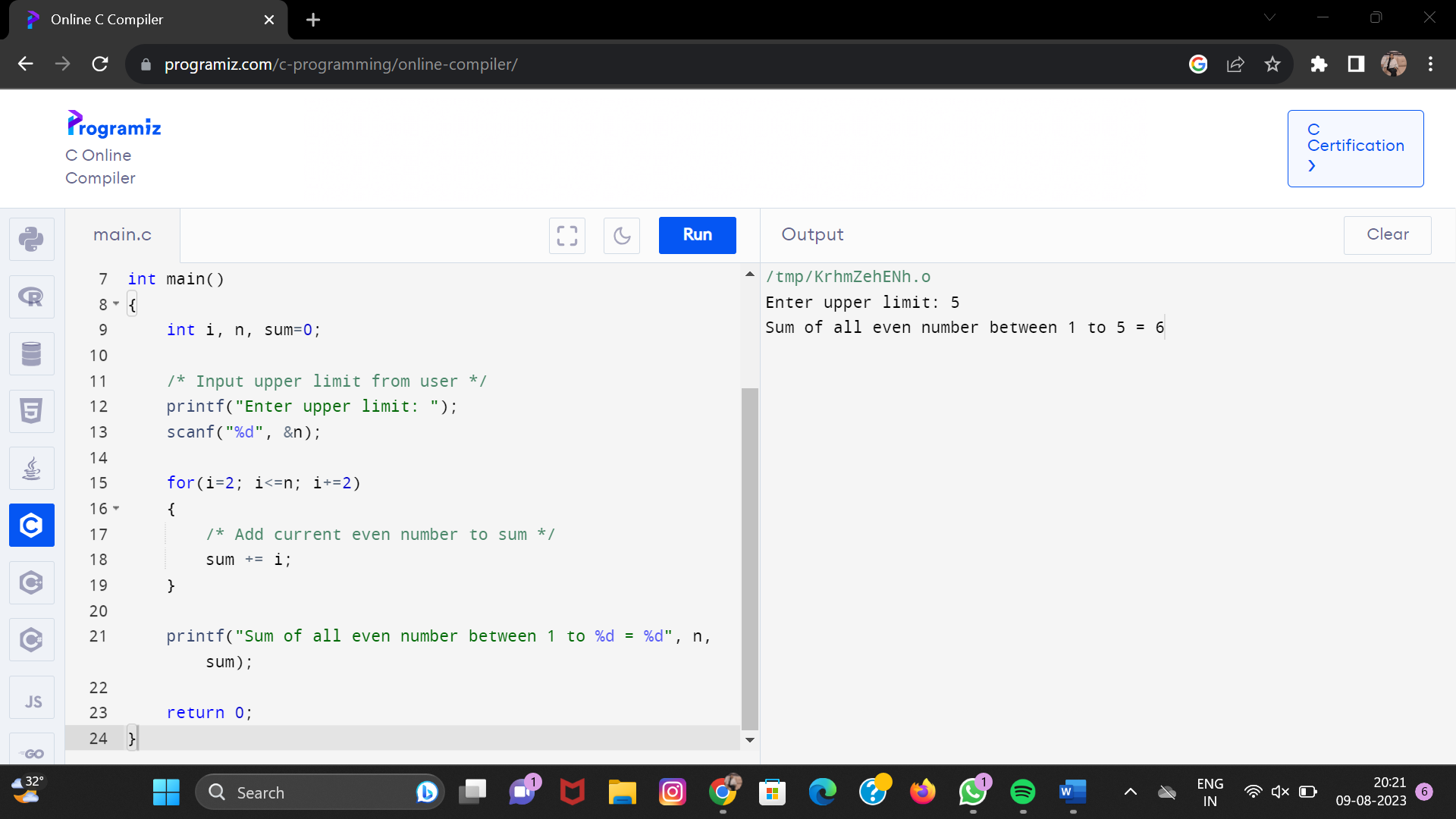
sum += i;

}

printf("Sum of all even number between 1 to %d = %d", n, sum);

return 0;

}



# *4.C program to reverse a number :*

#include <stdio.h>

int main()

{

int Num, rev\_Num = 0, remainder;

printf("Enter the number to reverse: ");

scanf("%d", &Num);

while (Num != 0)

{

remainder = Num % 10;

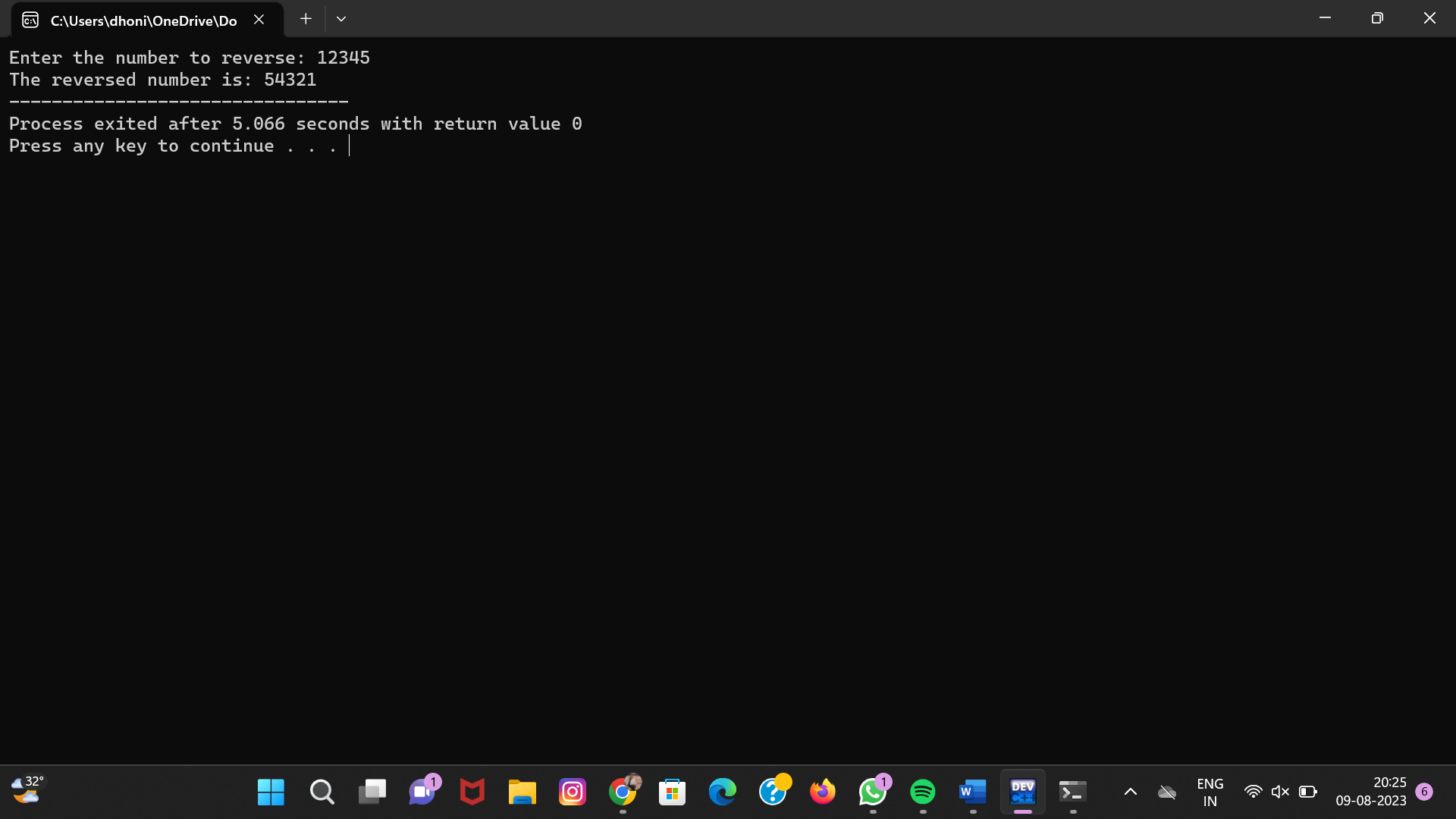
rev\_Num = rev\_Num\*10+ remainder;

Num = Num/10;

}

printf("The reversed number is: %d", rev\_Num);

return 0;

}

# *5.C program to check given number is palindrome:*

#include <stdio.h>

int main() {

int n, reversed = 0, remainder, original;

printf("Enter an integer: ");

scanf("%d", &n);

original = n;

while (n != 0) {

remainder = n % 10;

reversed = reversed \* 10 + remainder;

n /= 10;

}

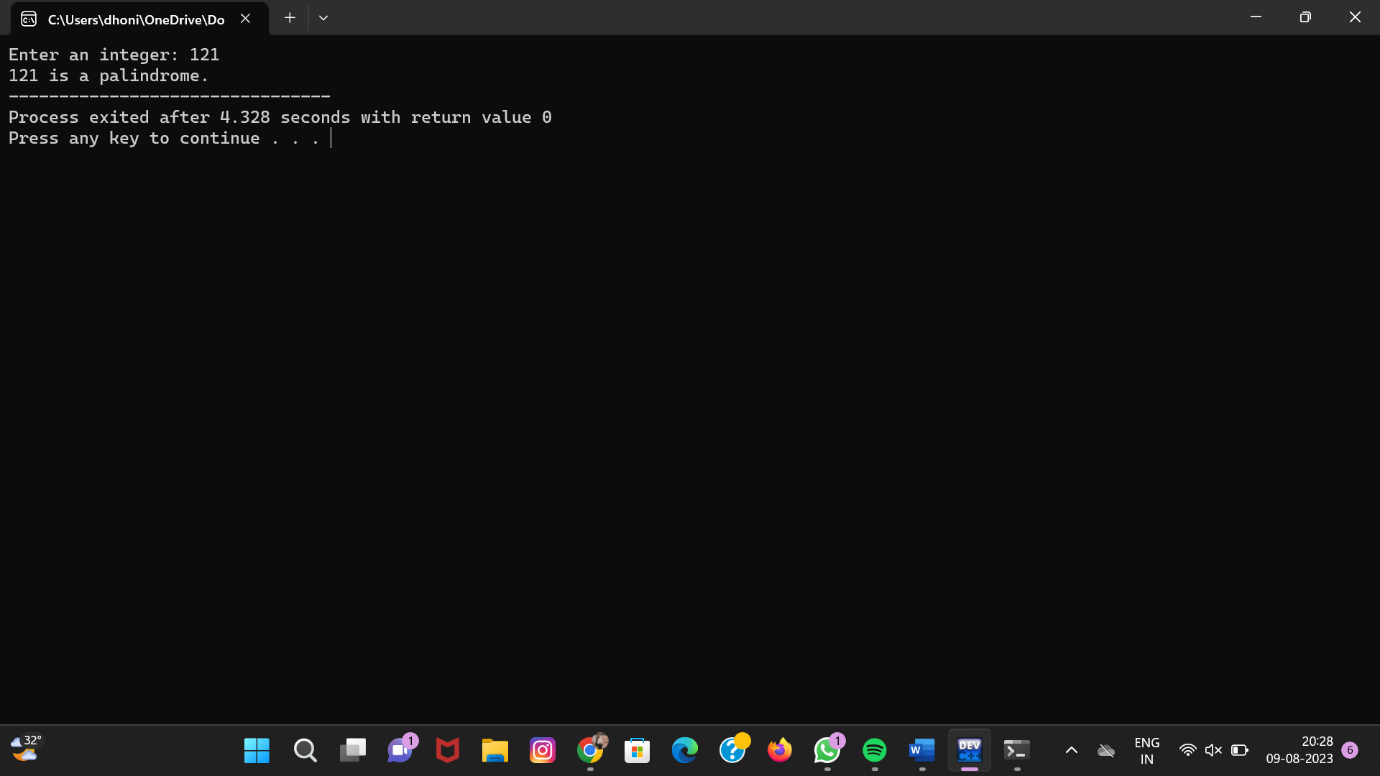
if (original == reversed)

printf("%d is a palindrome.", original);

else

printf("%d is not a palindrome.", original);

return 0;

}

# *6.C program to check whether the given number is Armstrong or not:*

#include <stdio.h>

int main() {

int num, originalNum, remainder, result = 0;

printf("Enter a three-digit integer: ");

scanf("%d", &num);

originalNum = num;

while (originalNum != 0) {

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

originalNum /= 10;

} if (result == num)

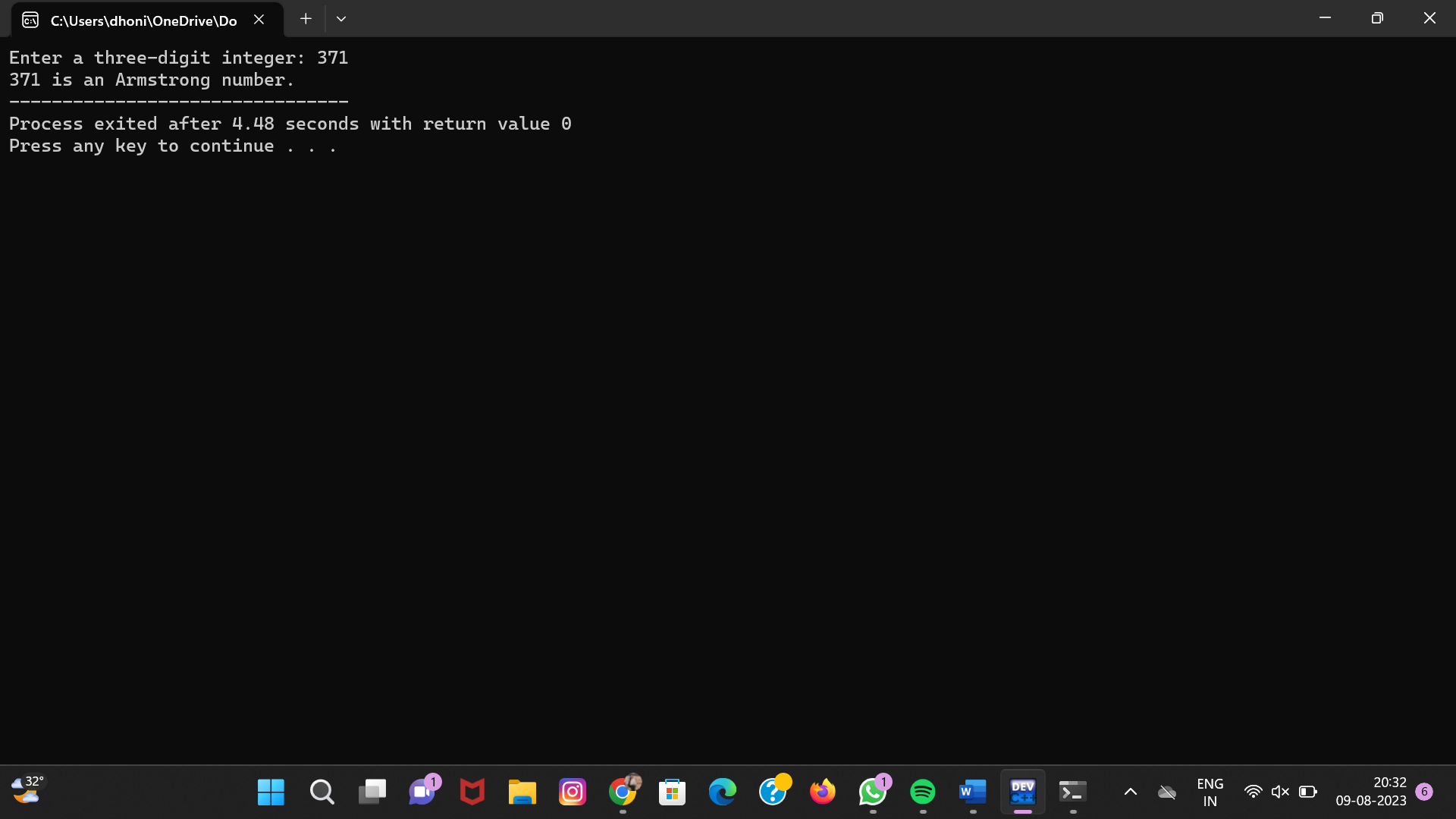
printf("%d is an Armstrong number.", num);

else

printf("%d is not an Armstrong number.", num);

return 0;

}



# *7.Find factorial of a given number without recursion:*

#include<stdio.h>

**int** main()

{

**int** i,fact=1,number;

 printf("Enter a number: ");

  scanf("%d",&number);

**for**(i=1;i<=number;i++){

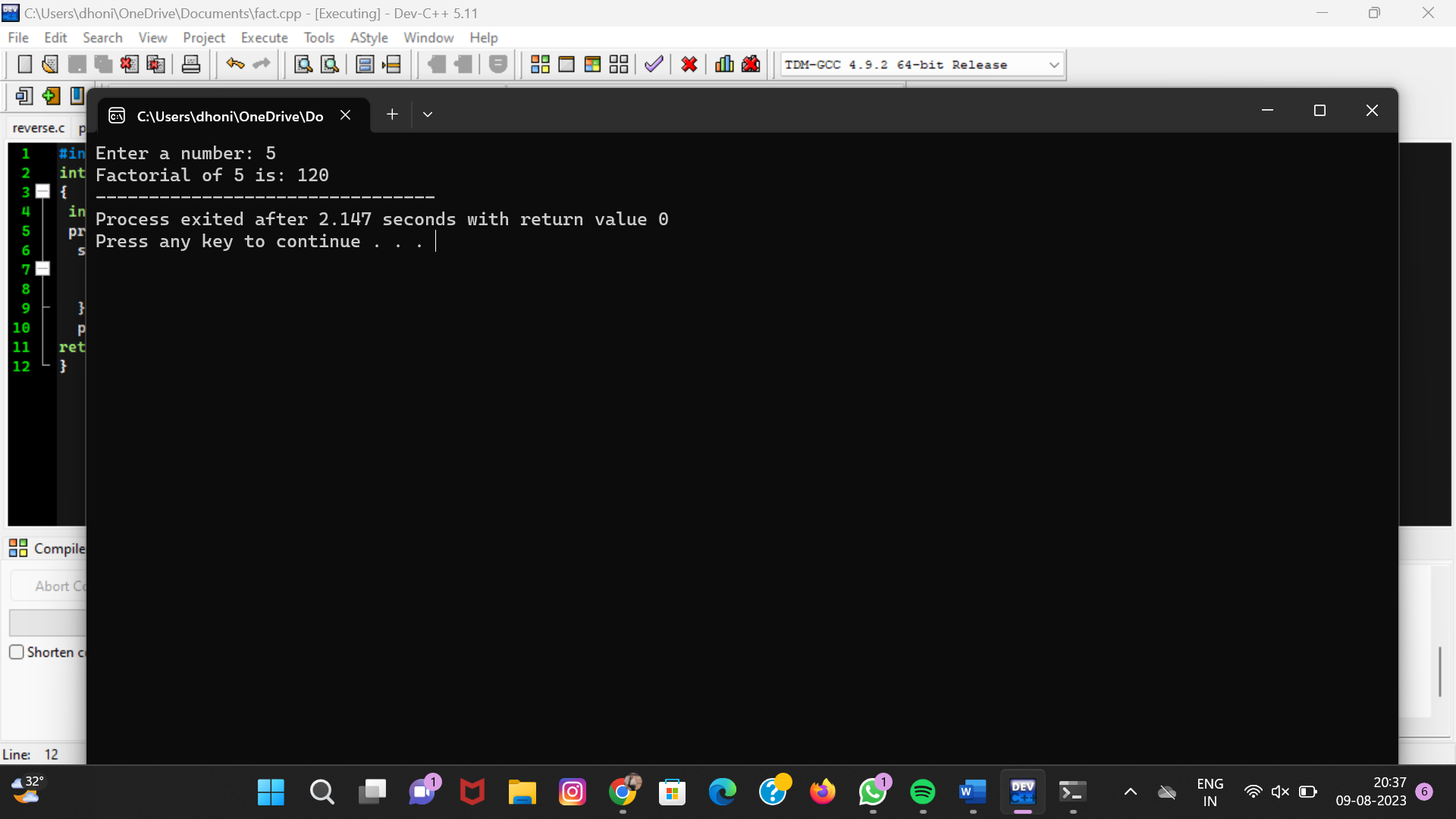
      fact=fact\*i;

  }

  printf("Factorial of %d is: %d",number,fact);

**return** 0;

}



# *8.Find factorial of a given number with recursion:*

#include<stdio.h>

**long** factorial(**int** n)

{

**if** (n == 0)

**return** 1;

**else**

**return**(n \* factorial(n-1));

}

**void** main()

{

**int** number;

**long** fact;

  printf("Enter a number: ");

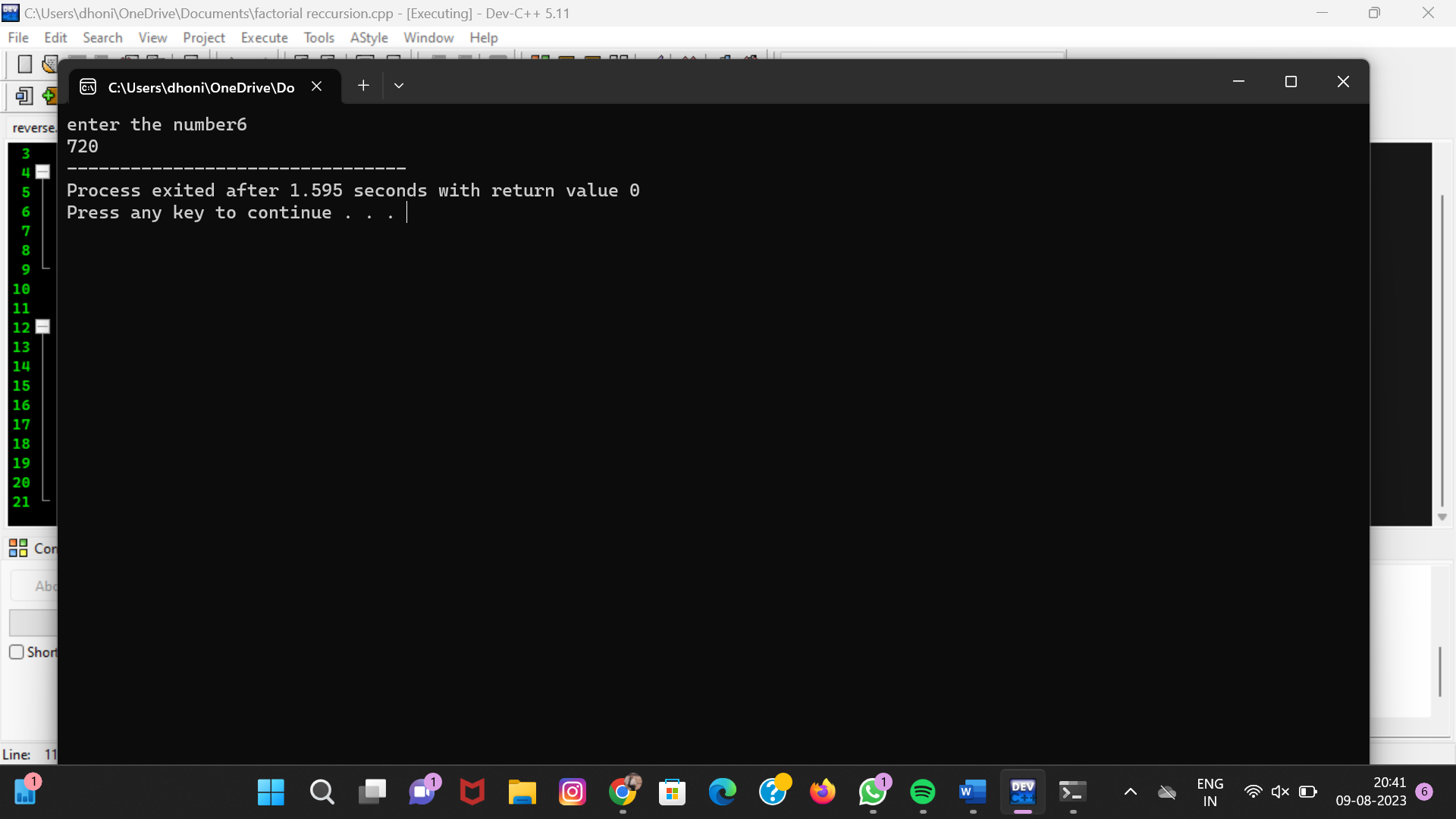
  scanf("%d", &number);

  fact = factorial(number);

  printf("Factorial of %d is %ld\n", number, fact);

**return** 0;

}



# *9.C program to generate fibonccie series with out recursion:*

#include<stdio.h>

**int** main()

{

**int** n1=0,n2=1,n3,i,number;

 printf("Enter the number of elements:");

 scanf("%d",&number);

 printf("\n%d %d",n1,n2”)

**for**(i=2;i<number;++i)

 {

  n3=n1+n2;

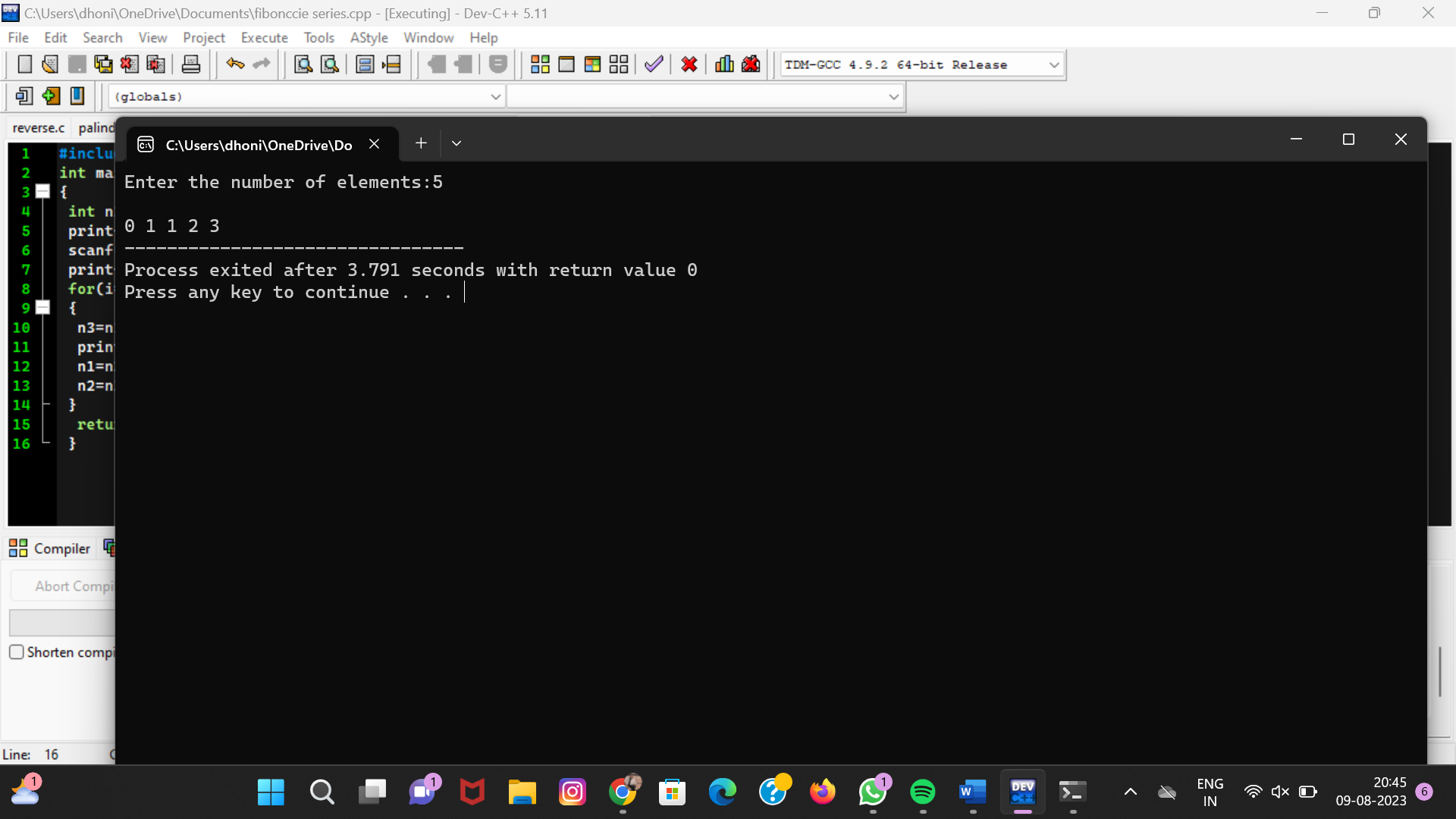
  printf(" %d",n3);

  n1=n2;

  n2=n3;

 }

**return** 0;

}    

# *10.C program to generate fibonccie series with recursion:*

#include<stdio.h>

**void** printFibonacci(**int** n){

**static** **int** n1=0,n2=1,n3;

**if**(n>0){

         n3 = n1 + n2;

         n1 = n2;

         n2 = n3;

         printf("%d ",n3);

         printFibonacci(n-1);

    }

}

**int** main(){

**int** n;

    printf("Enter the number of elements: ");

    scanf("%d",&n);

    printf("Fibonacci Series: ");

    printf("%d %d ",0,1);

    printFibonacci(n-2);//n-2 because 2 numbers are already printed

**return** 0;

 }

# 

# *11.To search a particular element in array using linear search:*

#include <stdio.h>

**int** linearSearch(**int** a[], **int** n, **int** val) {

  // Going through array sequencially

**for** (**int** i = 0; i < n; i++)

    {

**if** (a[i] == val)

**return** i+1;

    }

**return** -1;

}

**int** main() {

**int** a[] = {70, 40, 30, 11, 57, 41, 25, 14, 52}; // given array

**int** val = 41; // value to be searched

**int** n = **sizeof**(a) / **sizeof**(a[0]); // size of array

**int** res = linearSearch(a, n, val); // Store result

  printf("The elements of the array are - ");

**for** (**int** i = 0; i < n; i++)

  printf("%d ", a[i]);

  printf("\nElement to be searched is - %d", val);

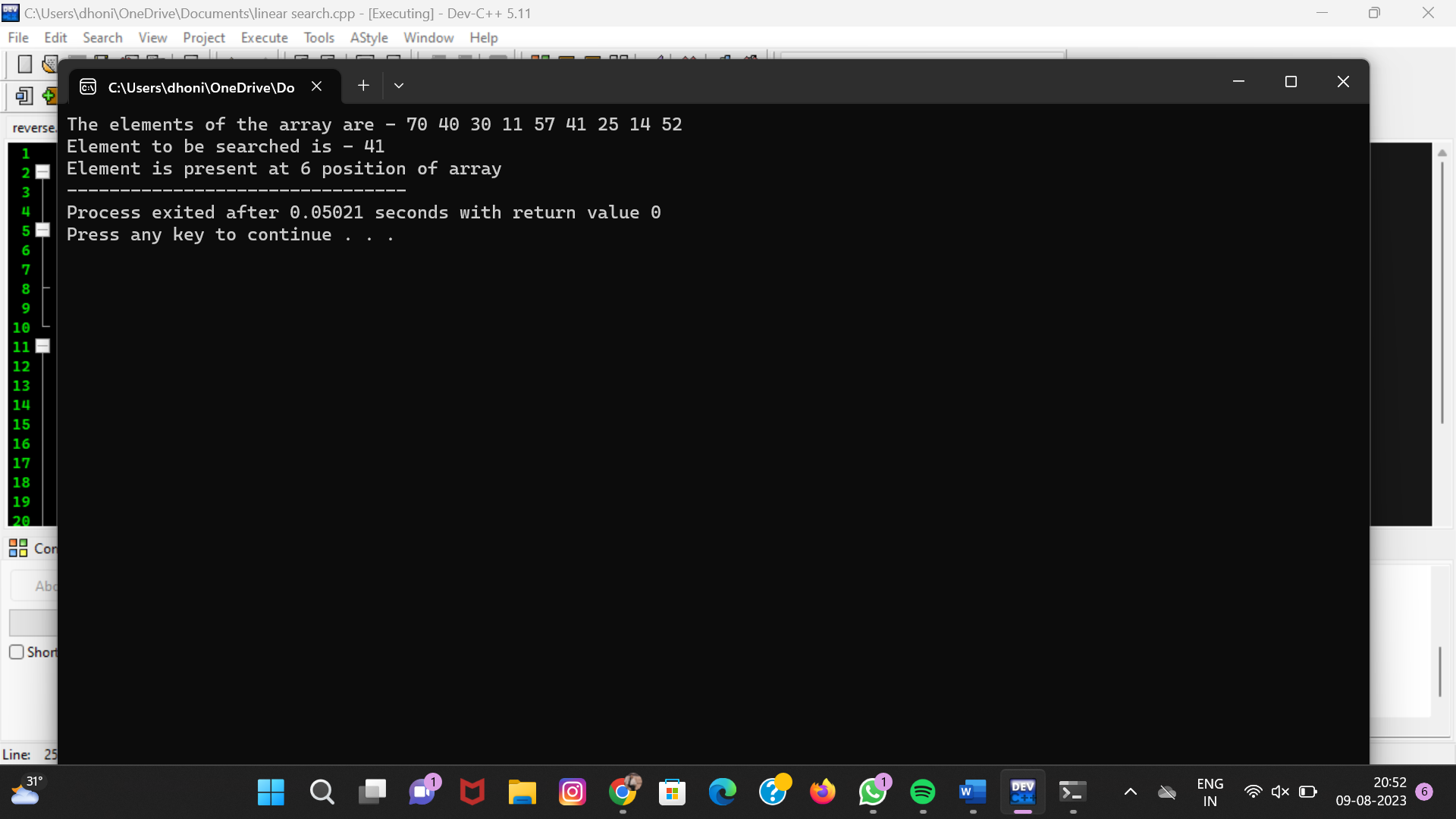
**if** (res == -1)

  printf("\nElement is not present in the array");

**else**

  printf("\nElement is present at %d position of array", res);

**return** 0;

}  

# *12.To search a number using binary search:*

#include <stdio.h>

int binarySearch(int arr[], int l, int r, int x)

{

if (r >= l)

{

int mid = l + (r - l)/2;

if (arr[mid] == x) return mid;

if (arr[mid] > x) return binarySearch(arr, l, mid-1, x);

return binarySearch(arr, mid+1, r, x);

}

return -1;

}

int main(void)

{

int arr[] = {2, 3, 4, 10, 40};

int n = sizeof(arr)/ sizeof(arr[0]);

int x = 10;

int result = binarySearch(arr, 0, n-1, x);

(result == -1)? printf("Element is not present in array");

printf("Element is present at index %d", result);

return 0;

}

