**Session 2 (unit 1): Library functions and scope & lifetime of variables (storage classes)**

|  |  |
| --- | --- |
| Roll No. A016 | Name: Varun Khadayate |
| Program: B-Tech (CSBS) | Division: SY |
| Batch: 1 | Date of Submission: 19-09-2020 |

1. **WAP to implement any 5 functions from string.h**

**ANS:**

* **strcat**

**CODE:** #include <cstring>

#include <iostream>

using namespace std;

int main()

{

char s1[] = "Varun";

char s2[] = " Khadayate";

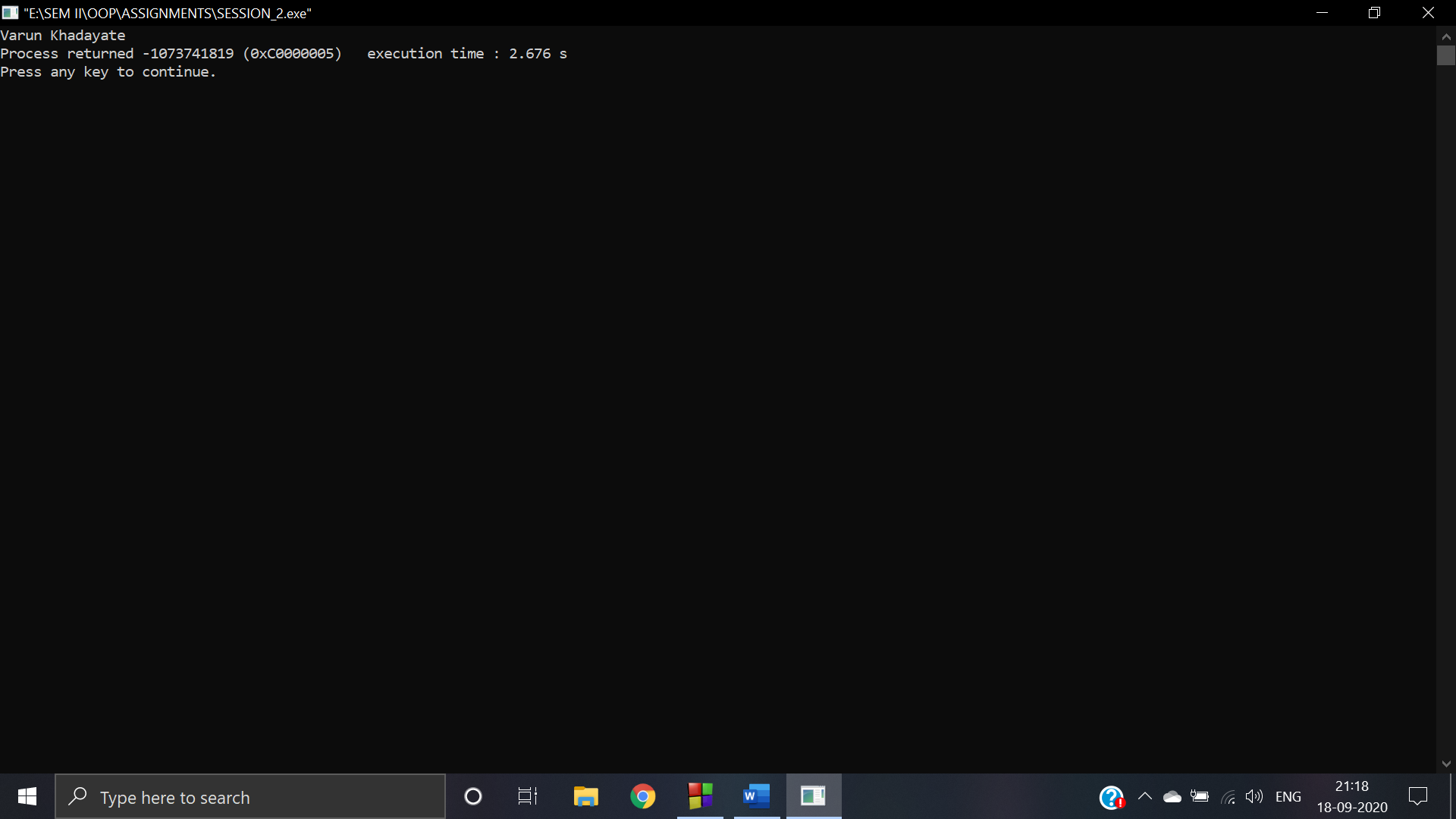
strcat(s1, s2);

cout << s1;

return 0;

}

**OUTPUT:**



* **strcmp**

**CODE:** #include <cstring>

#include <iostream>

using namespace std;

int main()

{

char s1[] = "Varun";

char s2[] = "Khadayate";

int x = strcmp(s1, s2);

if(x == 0)

{

cout<< "The both strings are similar"<<endl;

}

else

{

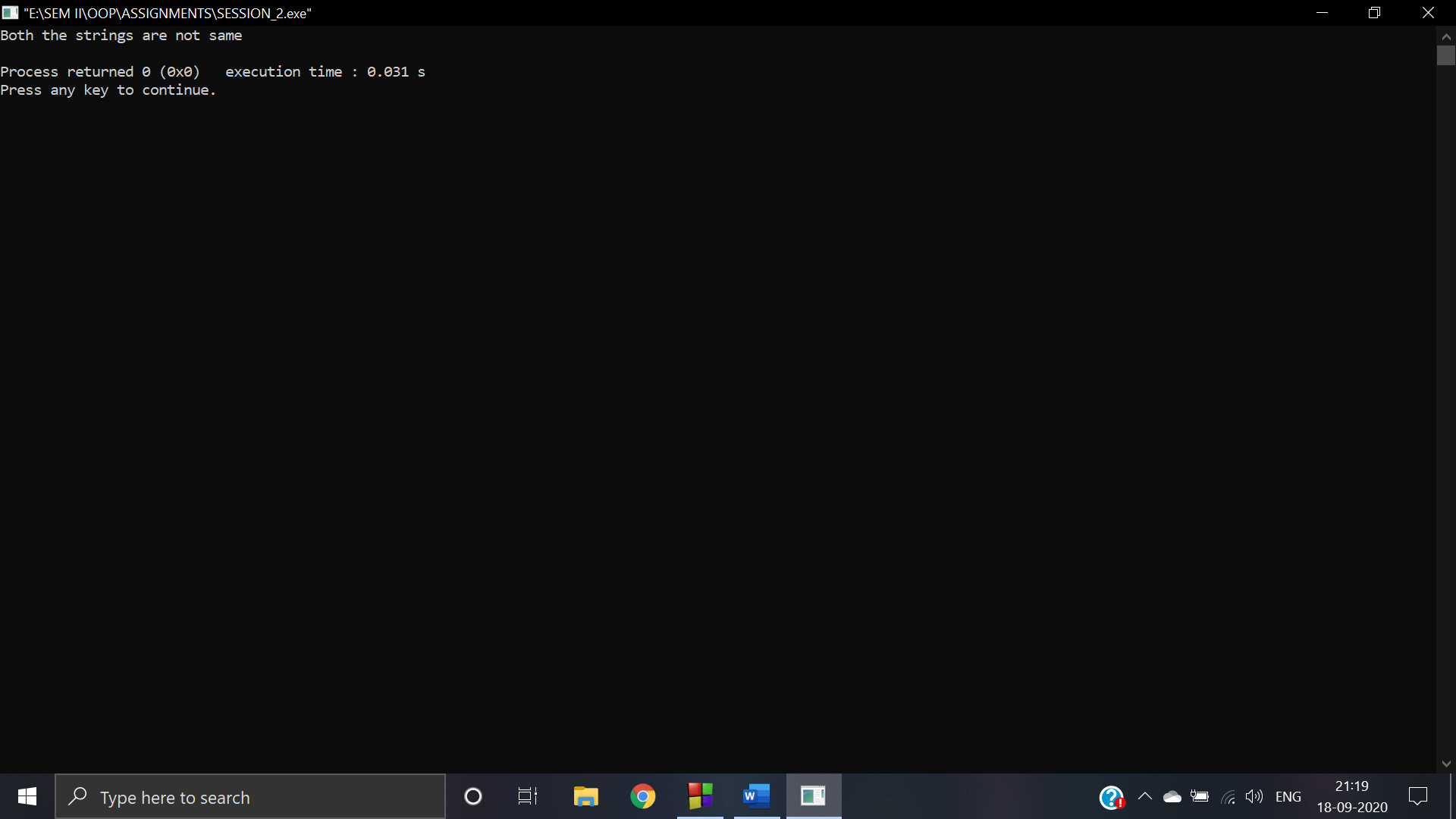
cout<<"Both the strings are not same"<<endl;

}

return 0;

}

**OUTPUT:**



* **strlen**

**CODE:** #include <cstring>

#include <iostream>

using namespace std;

int main()

{

char s1[] = "Varun Khadayate";

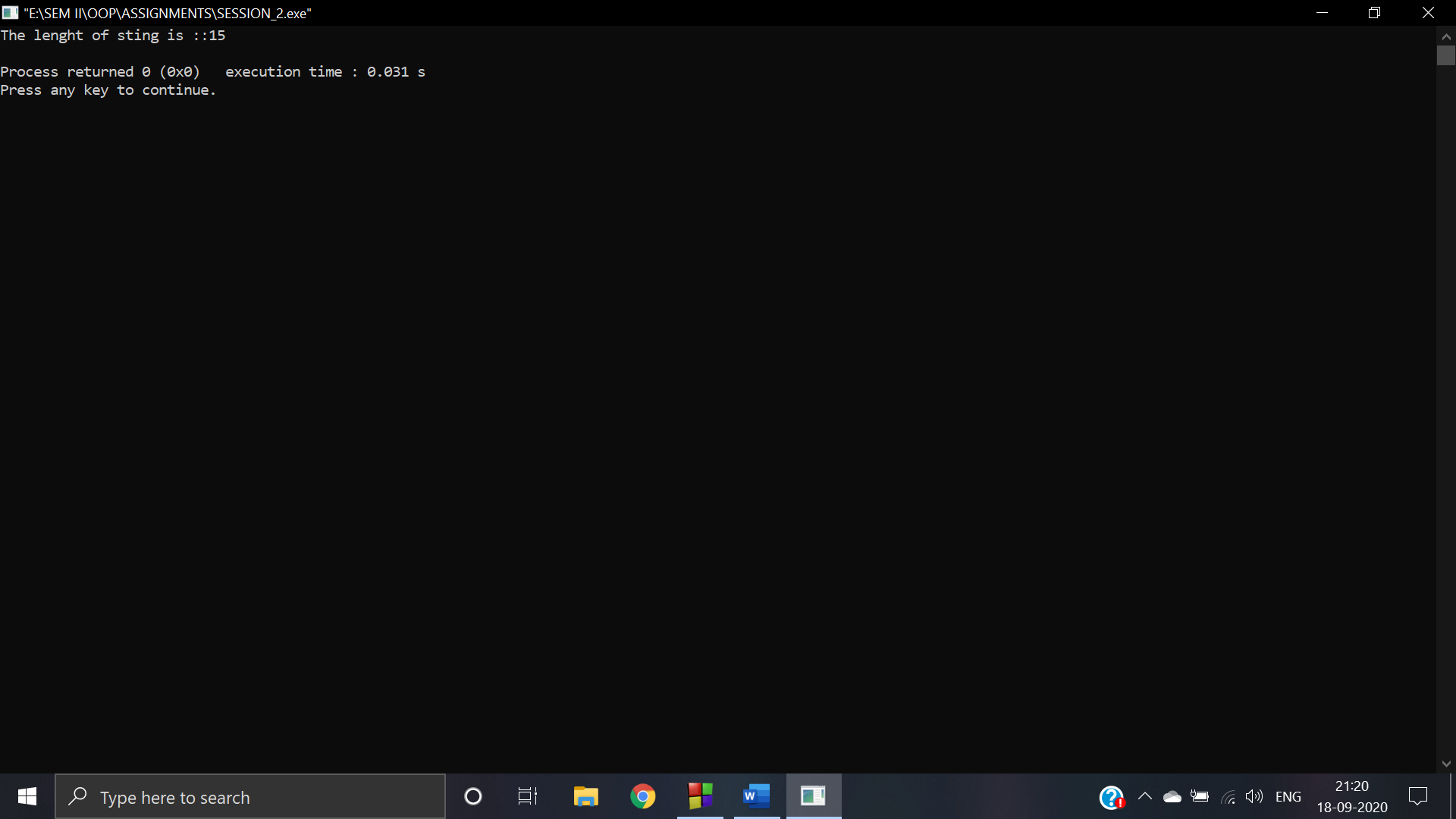
int x = strlen(s1);

cout<<"The lenght of sting is ::"<<x<<endl;

return 0;

}

**OUTPUT:**



* **strncat**

**CODE:** #include <cstring>

#include <iostream>

using namespace std;

int main()

{

char s1[] = "Varun";

char s2[] = " Khadayate";

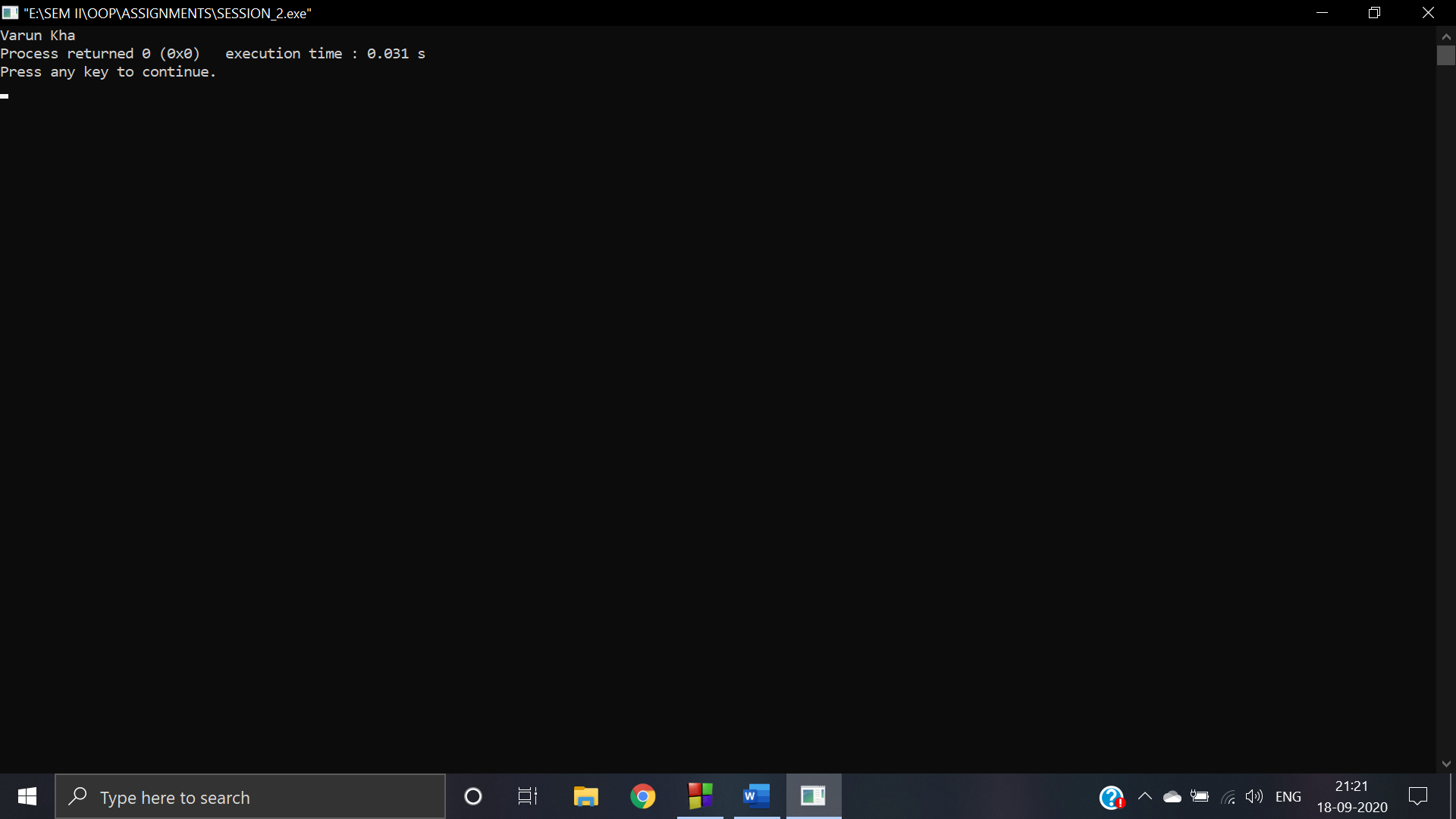
strncat(s1, s2,4);

cout << s1;

return 0;

}

**OUTPUT:**



* **strnlen**

**CODE:** #include <cstring>

#include <iostream>

using namespace std;

int main()

{

char s1[] = "Varun Khadayate";

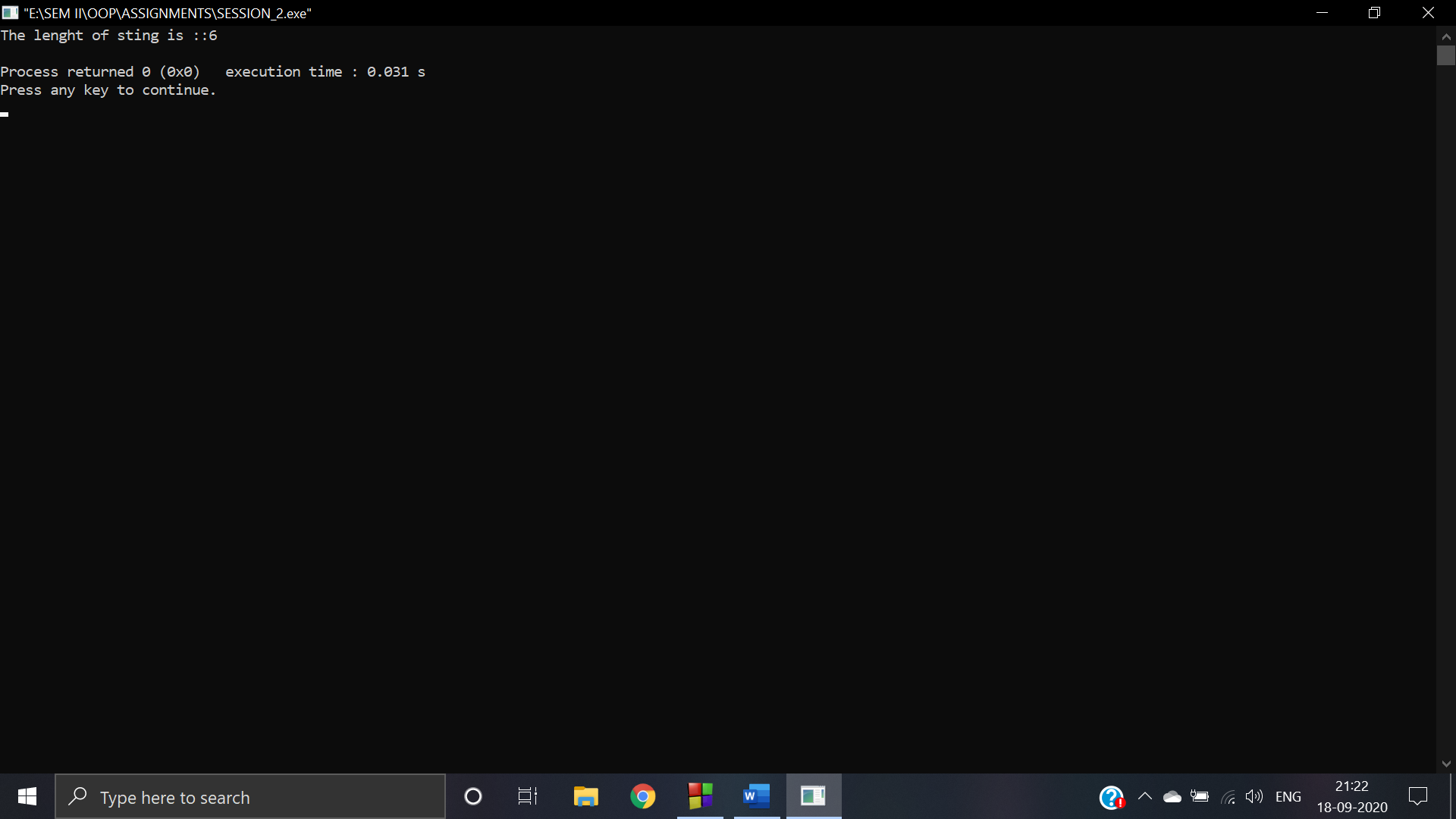
int x = strnlen(s1, 6);

cout<<"The lenght of sting is ::"<<x<<endl;

return 0;

}

**OUTPUT:**



1. **WAP to implement any 5 functions from math.h**

**ANS:**

* **double sin(double)**

**CODE:** #include <iostream>

#include <math.h>

using namespace std;

int main()

{

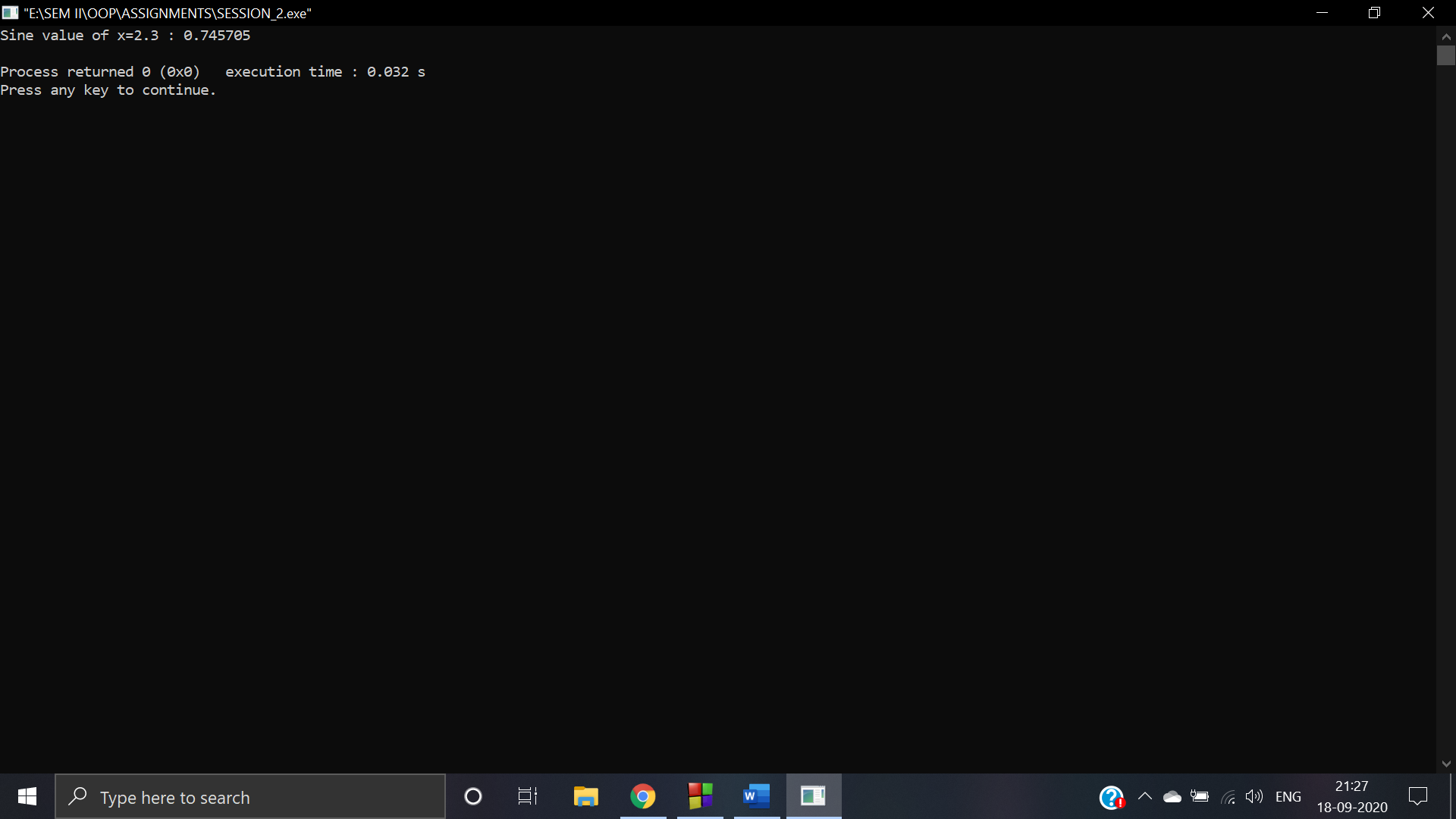
double x = 2.3;

cout << "Sine value of x=2.3 : " << sin(x) << endl;

return 0;

}

**OUTPUT:**



* **double sqrt(double)**

**CODE:** #include <iostream>

#include <math.h>

using namespace std;

int main()

{

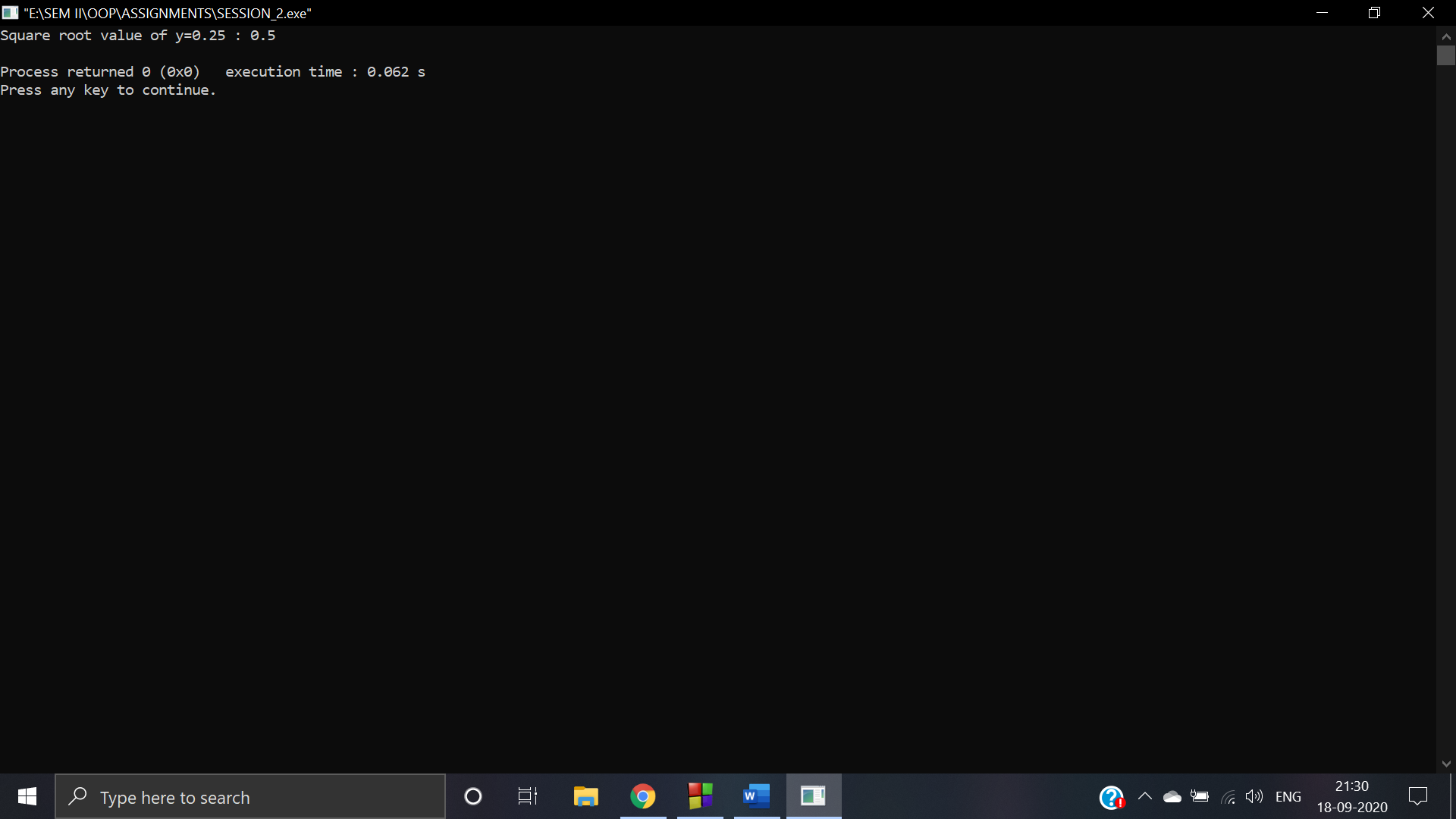
double y = 0.25;

cout << "Square root value of y=0.25 : " << sqrt(y) << endl;

return 0;

}

**OUTPUT:**



* **int abs(int)**

**CODE:** #include <iostream>

#include <math.h>

using namespace std;

int main()

{

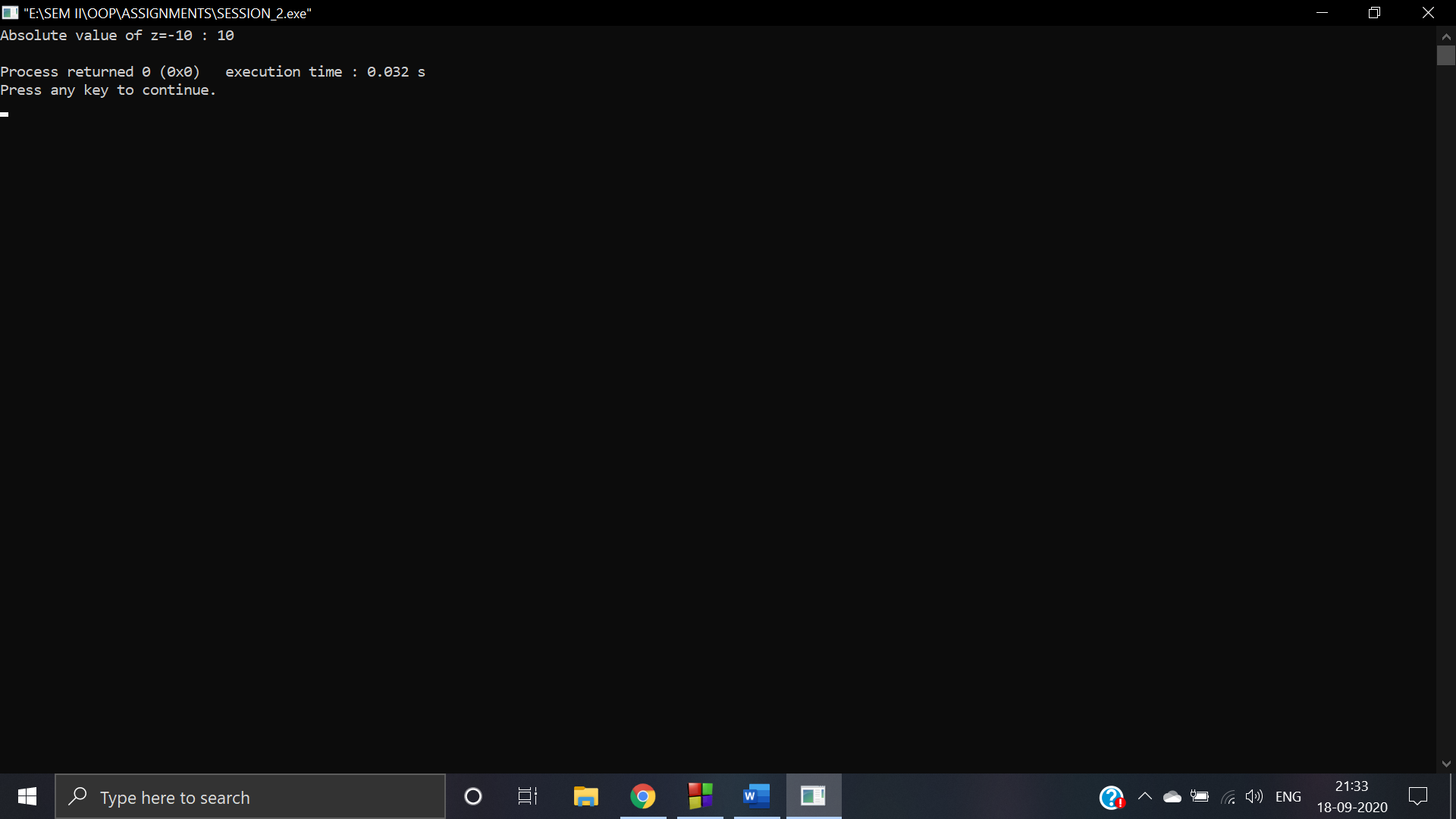
int z = -10;

cout << "Absolute value of z=-10 : " << abs(z) << endl;

return 0;

}

**OUTPUT:**



* **double pow(double, double)**

**CODE:** #include <iostream>

#include <math.h>

using namespace std;

int main()

{

double x = 2.3;

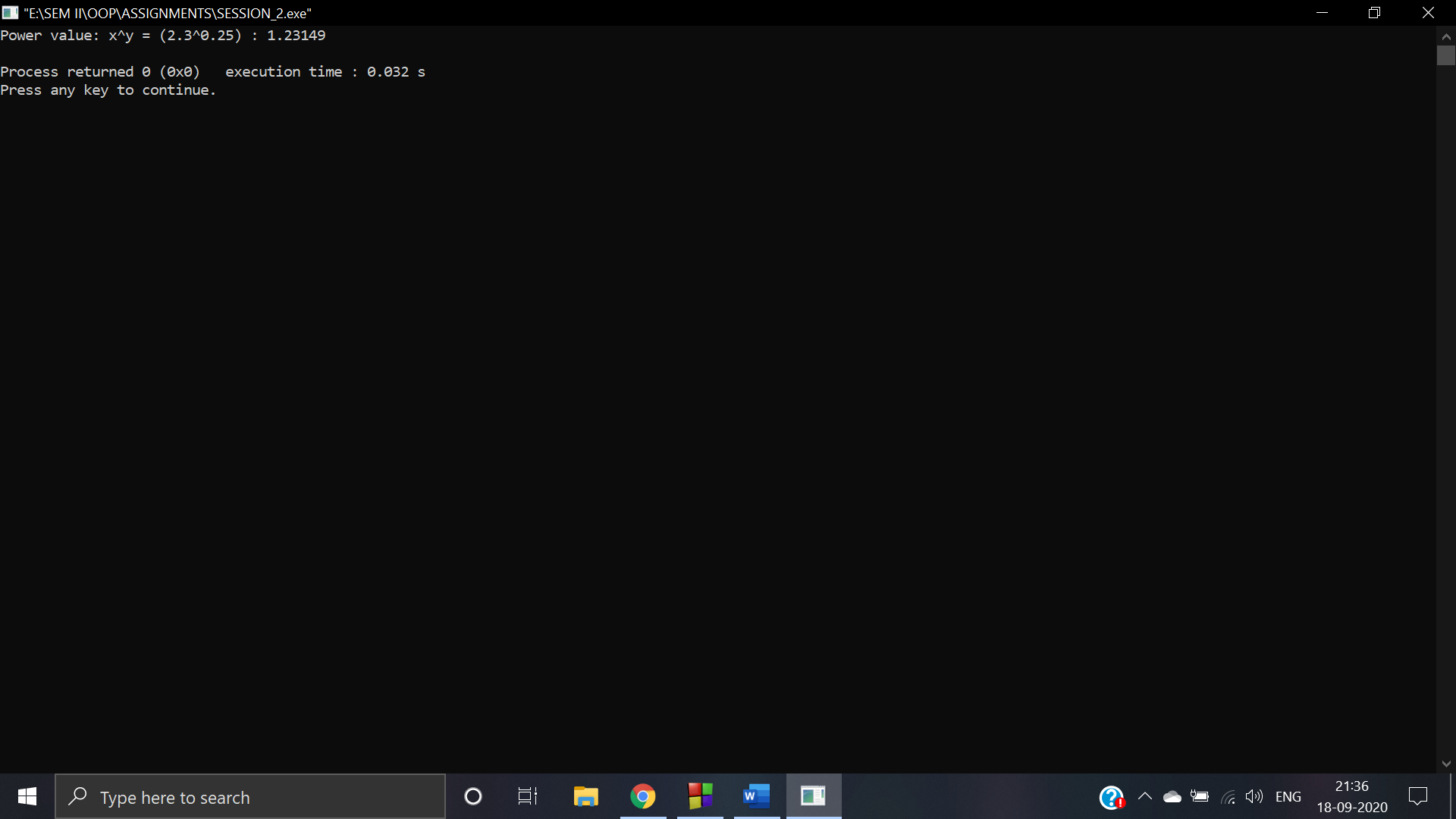
double y = 0.25;

cout << "Power value: x^y = (2.3^0.25) : " << pow(x, y) << endl;

return 0;

}

**OUTPUT:**



* **double fabs(double)**

**CODE:** #include <iostream>

#include <math.h>

using namespace std;

int main()

{

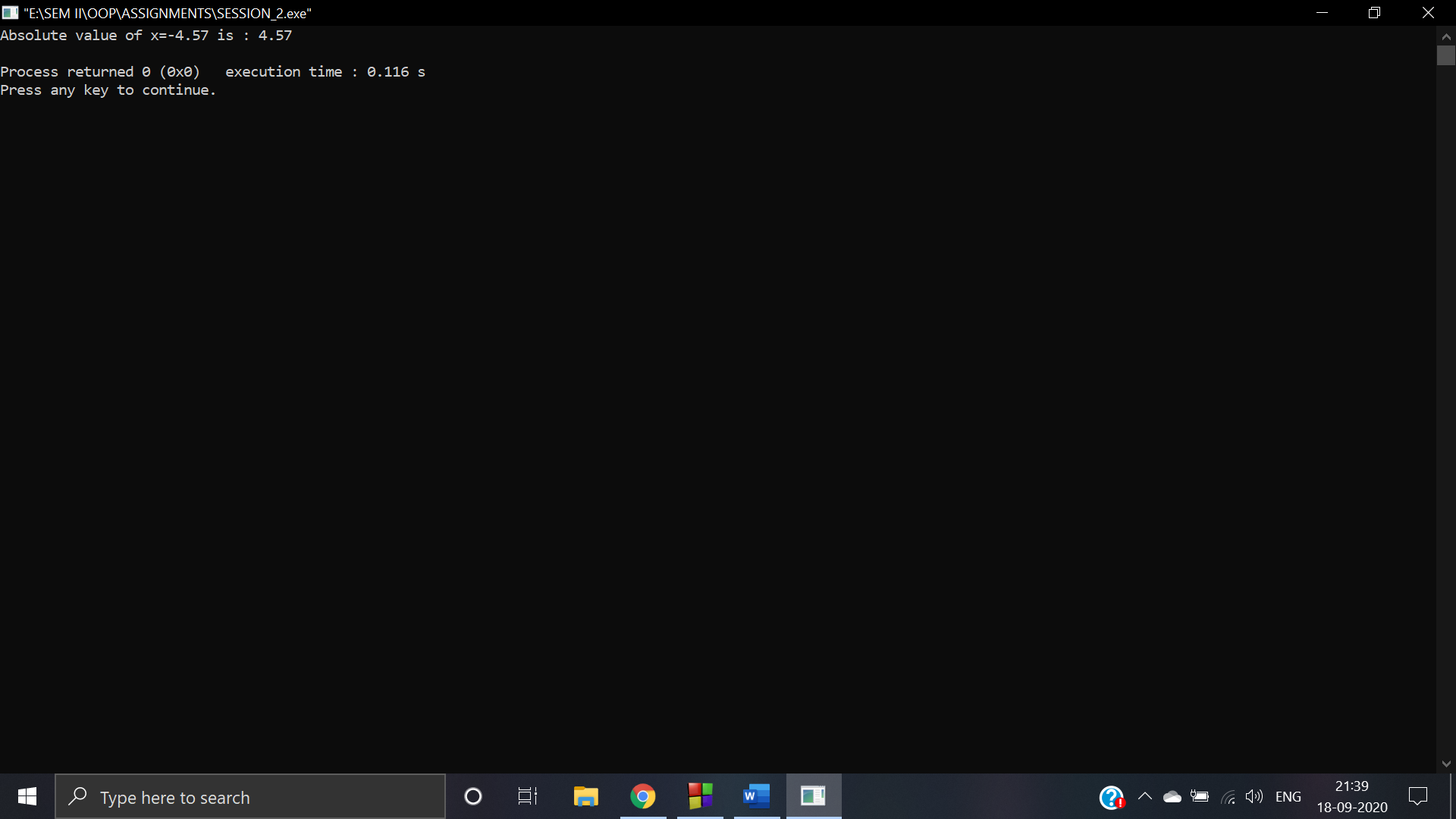
double x = -4.57;

cout << "Absolute value of x=-4.57 is : " << fabs(x) << endl;

return 0;

}

**OUTPUT:**



1. **WAP depicting working of storage classes in C++ (Automatic, External, Static, and Register)**

**ANS:**

* **AUTOMATIC**

**CODE:** #include<iostream>

#include <stdio.h>

using namespace std;

void sum(int n1, int n2)

{

auto sum = n1+n2;

cout<<"sum is : "<<sum;

}

int main()

{

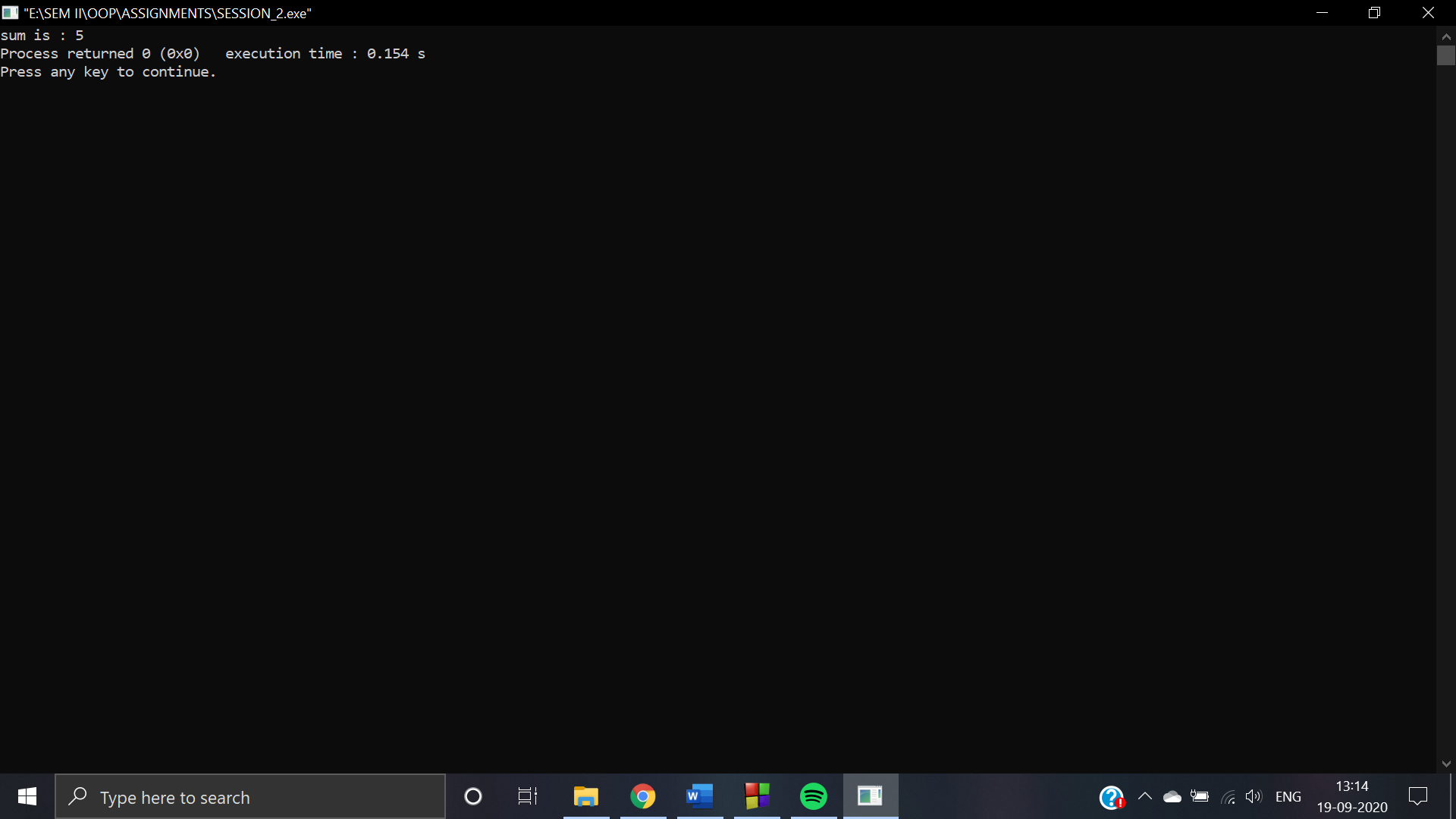
int i = 2, j = 3, k;

sum(i, j);

return 0;

}

**OUTPUT:**



* **EXTERNAL**

**CODE:** #include <iostream>

using namespace std;

int x;

void externStorageClass()

{

extern int x;

cout << "Value of the variable 'x' declared as extern: " << x << "\n";

x = 2;

cout << "Modified value of the variable 'x' declared as extern: "<< x;

}

int main()

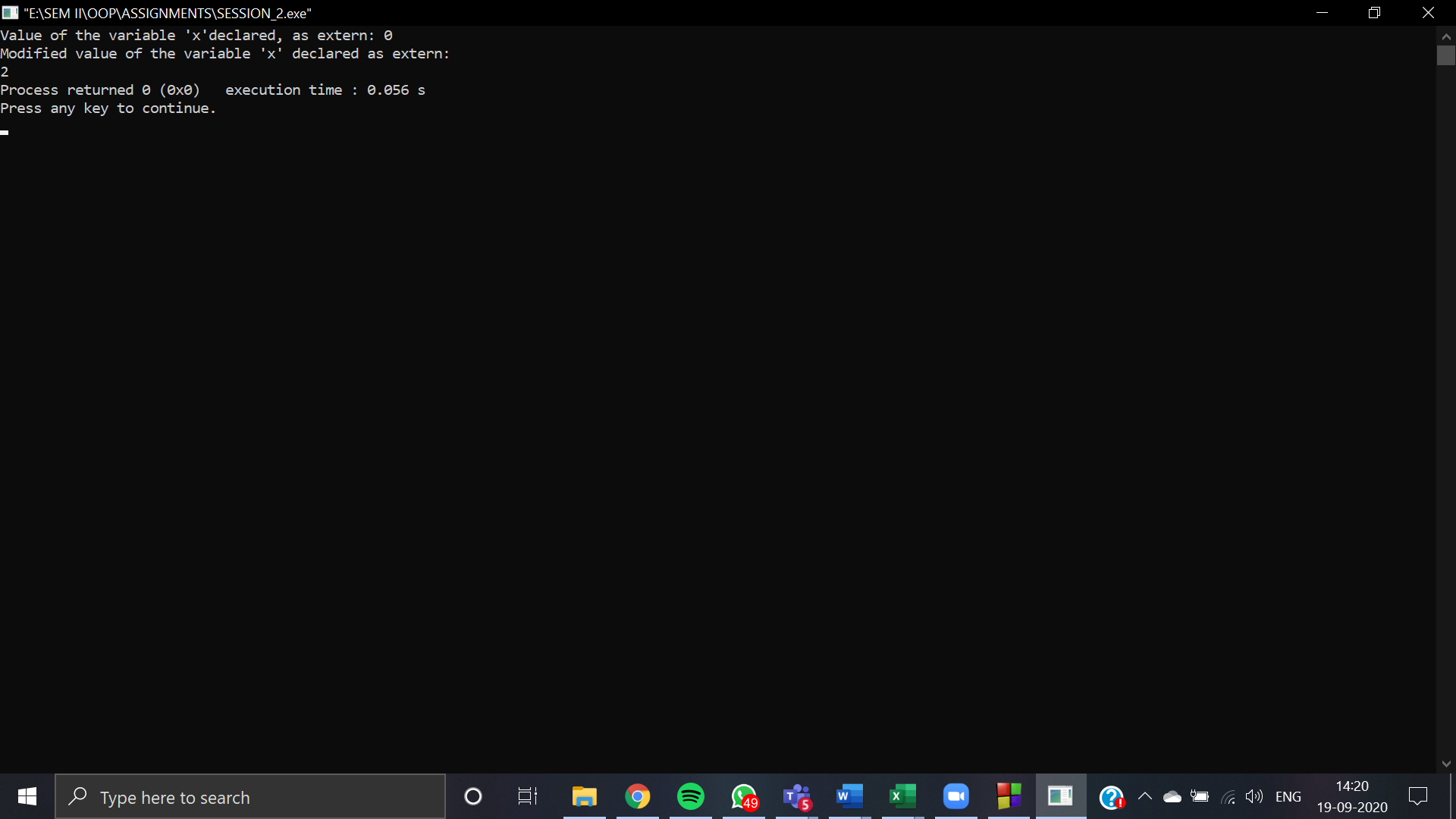
{

externStorageClass();

return 0;

}

**OUTPUT:**



* **STATIC**

**CODE:** #include <iostream>

using namespace std;

int staticNum()

{

cout << "For static variables: ";

static int count = 0;

count++;

return count;

}

int main()

{

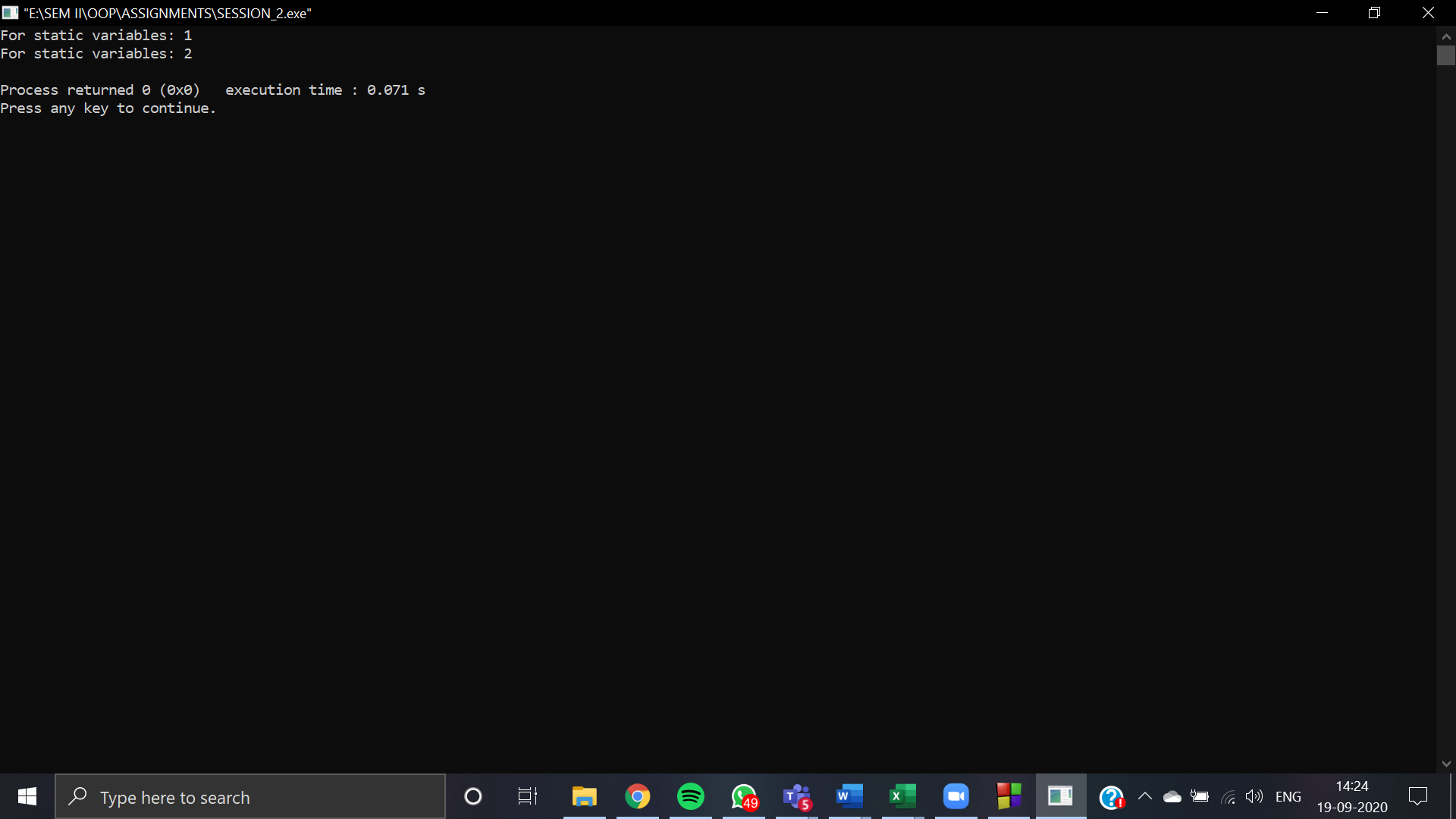
cout << staticNum() << "\n";

cout << staticNum() << "\n";

return 0;

}

**OUTPUT:**



* **REGISTER**

**CODE:** #include <iostream>

using namespace std;

void registerStorageClass()

{

cout << "Demonstrating register class\n";

register char b = 'G';

cout << "Value of the variable 'b' declared as register: " << b;

}

int main()

{

registerStorageClass();

return 0;

}

**OUTPUT:**

