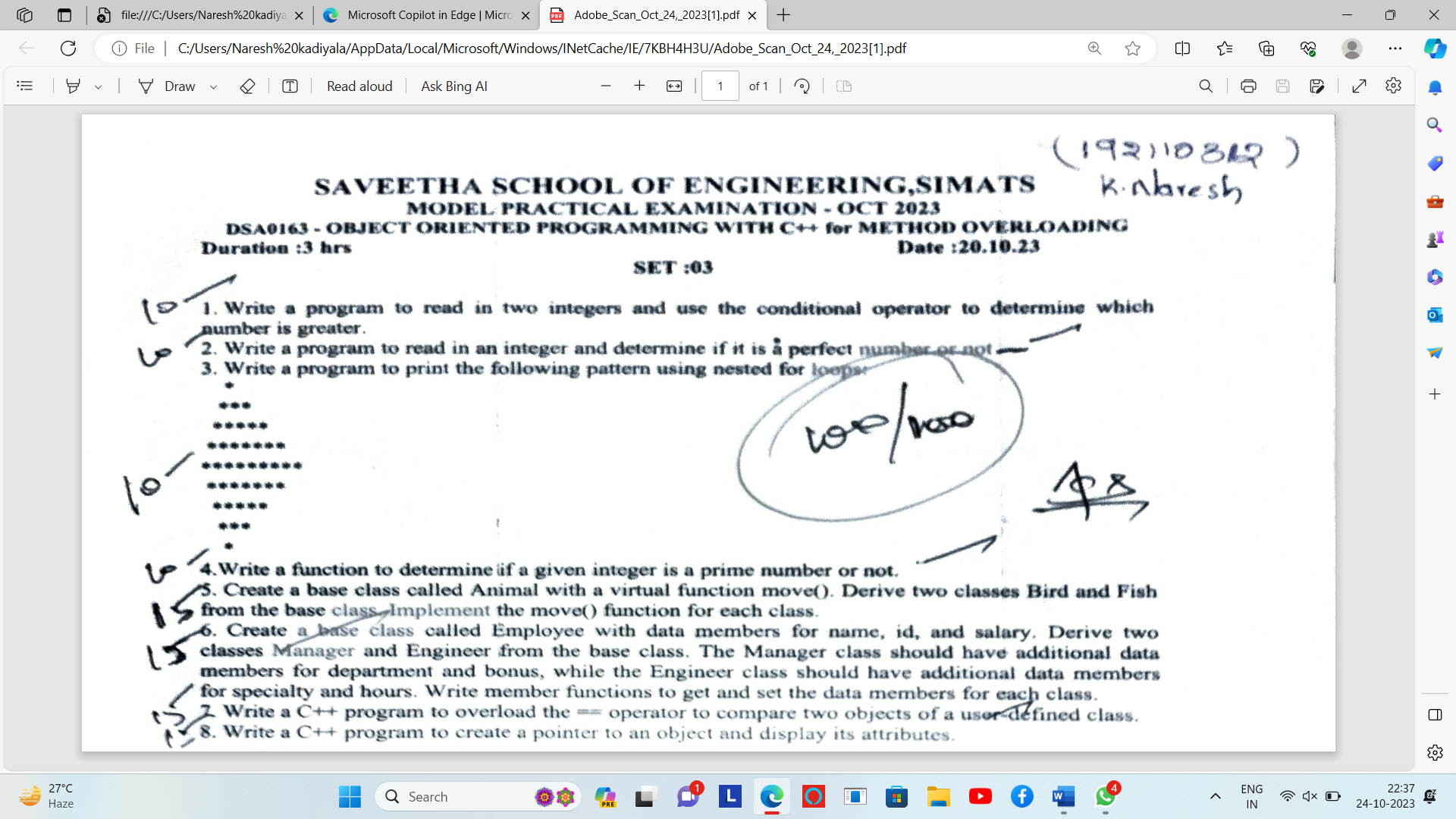
**Model exam**

Name: KadiyalaNaresh

Rg.no.: 192110312

Course: object oriented C++

Code: DSA0163

1. 

Code

#include <iostream>

using namespace std;

int main() {

int num1, num2;

cout << "Enter the first integer: ";

cin >> num1;

cout << "Enter the second integer: ";

cin >> num2;

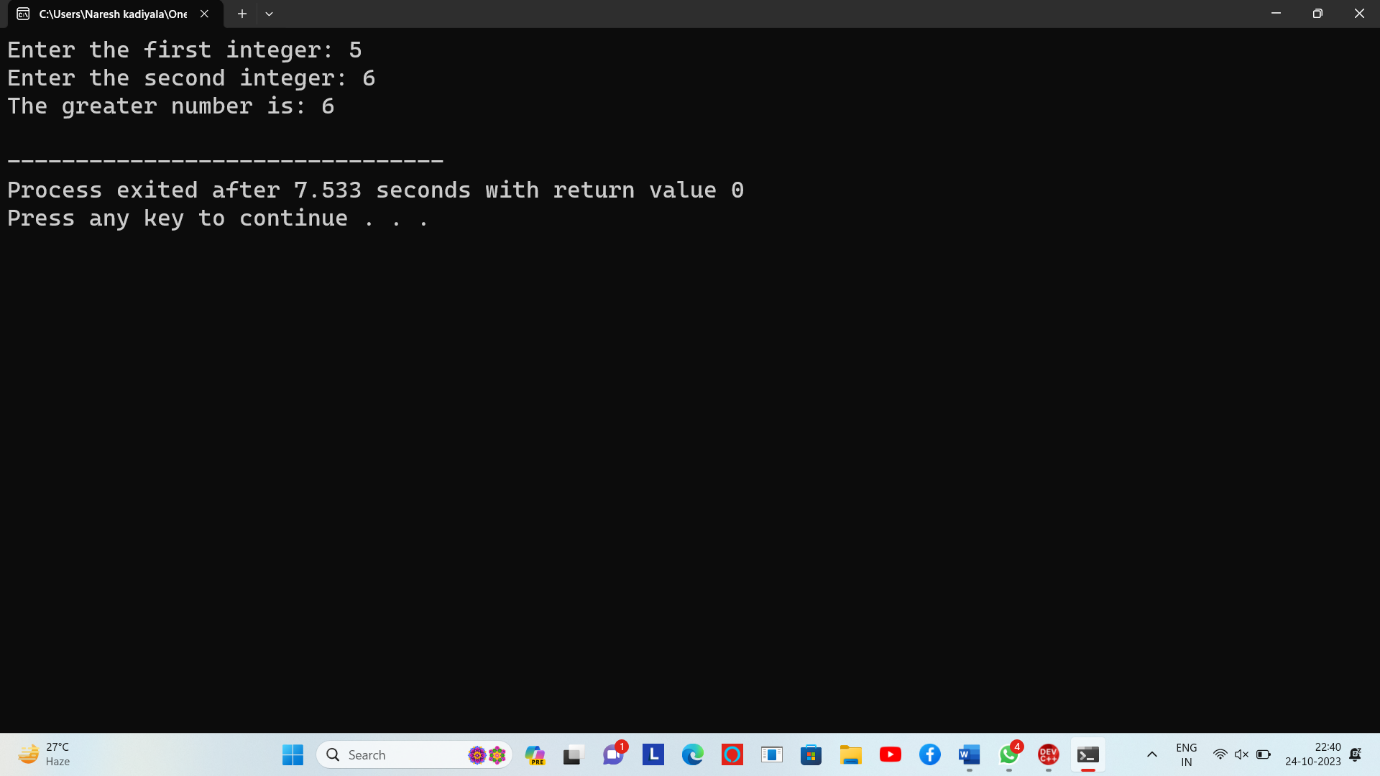
int greater = (num1 > num2) ? num1 : num2;

cout << "The greater number is: " << greater <<endl;

return 0;

}

**Output:-**

****

**2.)code**

#include <iostream>

using namespace std;

int main ()

{

int num , sum = 0;

int i = 1;

cout<<"Enter the number: ";

cin>>num;

while(i <= num/2)

{

if(num % i == 0)

sum = sum + i;

i++;

}

if(sum == num)

cout << num << " is a perfect number";

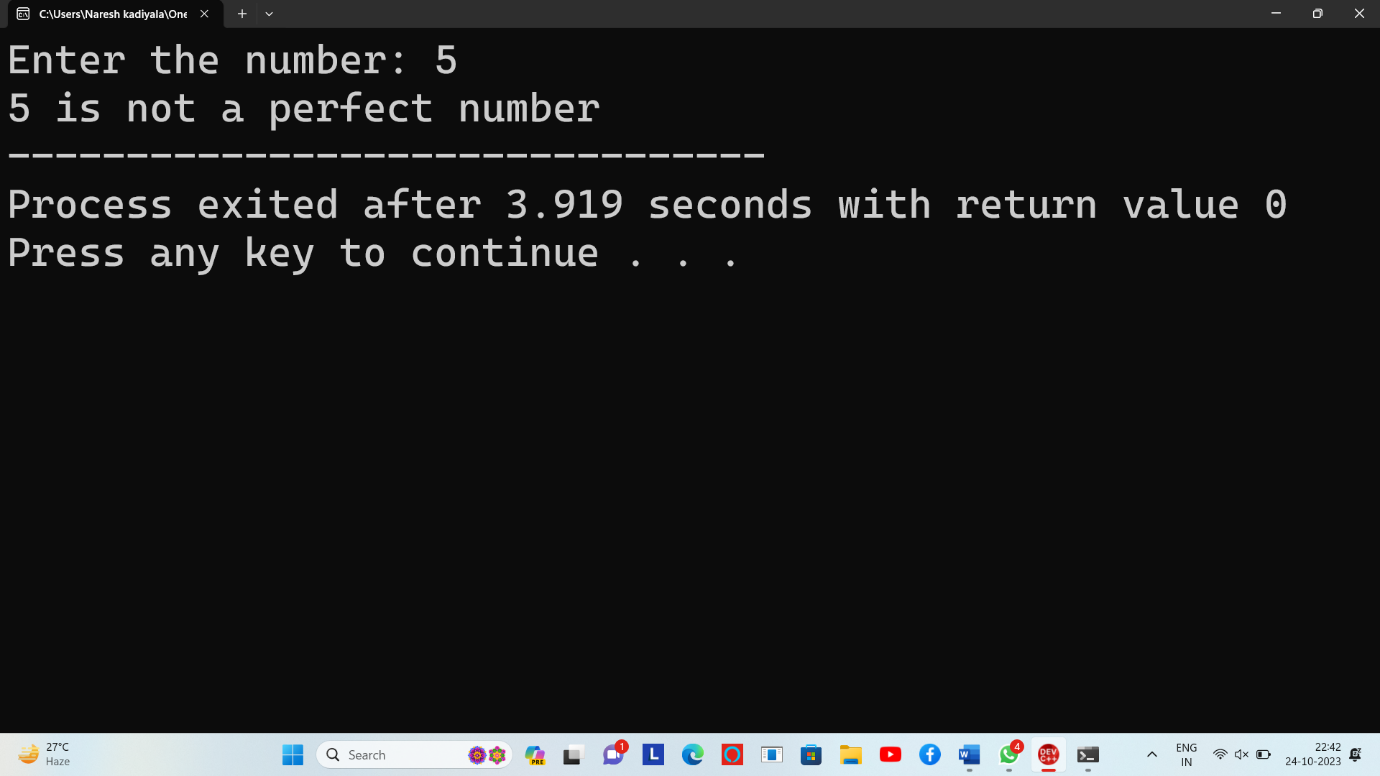
else

cout << num << " is not a perfect number";

return 0;

}

**Output:-**

****

**3.)code**

#include <iostream>

using namespace std;

int main()

{

int n, s, x, y;

cout << "Enter number of rows: ";

cin >> n;

for(x = 0; x <= n; x++)

{

for(s = n; s > x; s--)

cout << " ";

for(y = 0; y < x; y++)

cout << "\* ";

cout << "\n";

}

for(x = 1; x < n; x++)

{

for(s = 0; s < x; s++)

cout << " ";

for(y = n; y > x; y--)

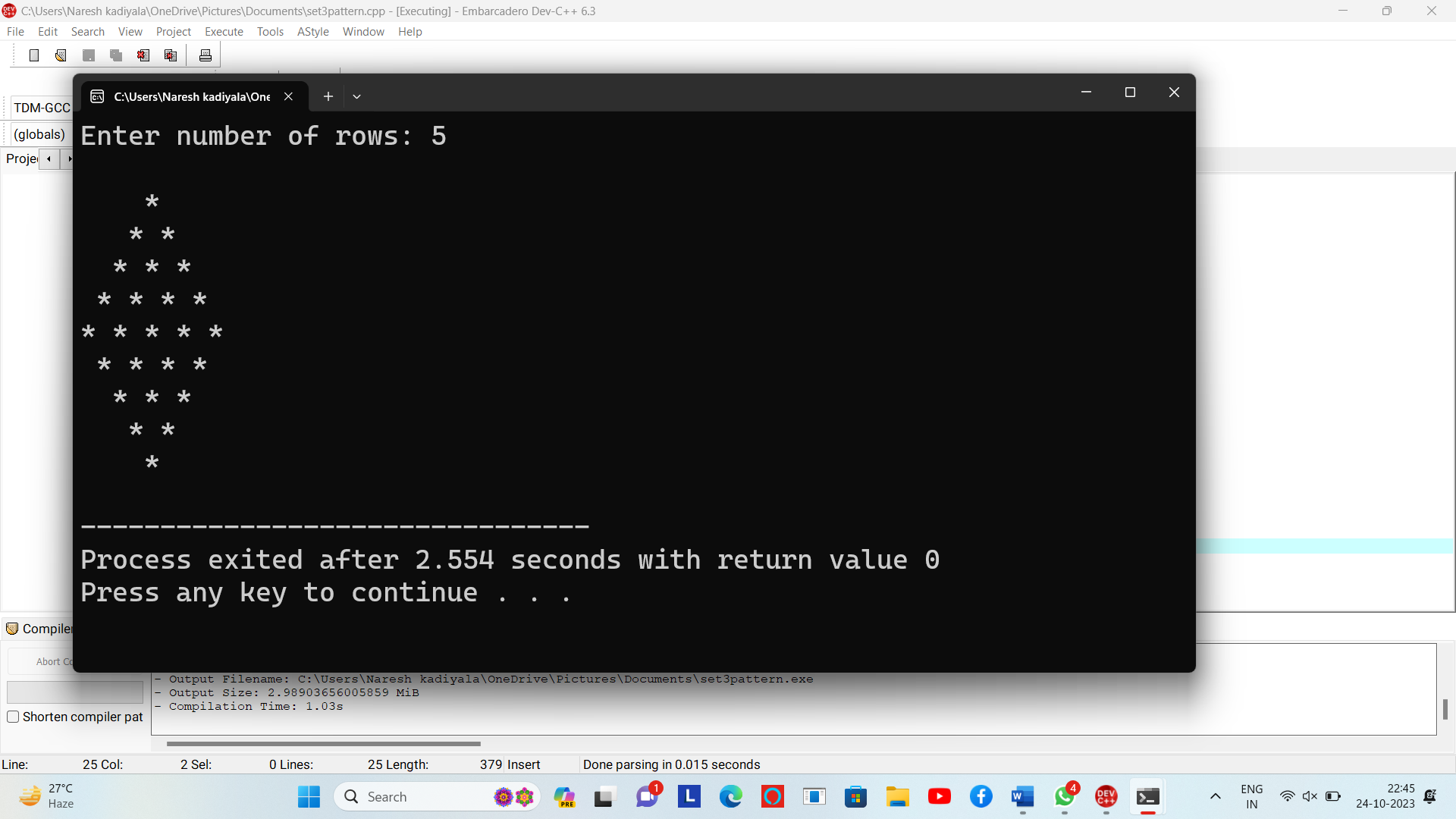
cout << "\* ";

cout << "\n";

}

return 0;

}

**Output:-**

**4.)code**

#include <iostream>

using namespace std;

int main() {

int i, n;

bool is\_prime = true;

cout << "Enter a positive integer: ";

cin >> n;

if (n == 0 || n == 1) {

is\_prime = false;

}

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

if (n % i == 0) {

is\_prime = false;

break;

}

}

if (is\_prime)

cout << n << " is a prime number";

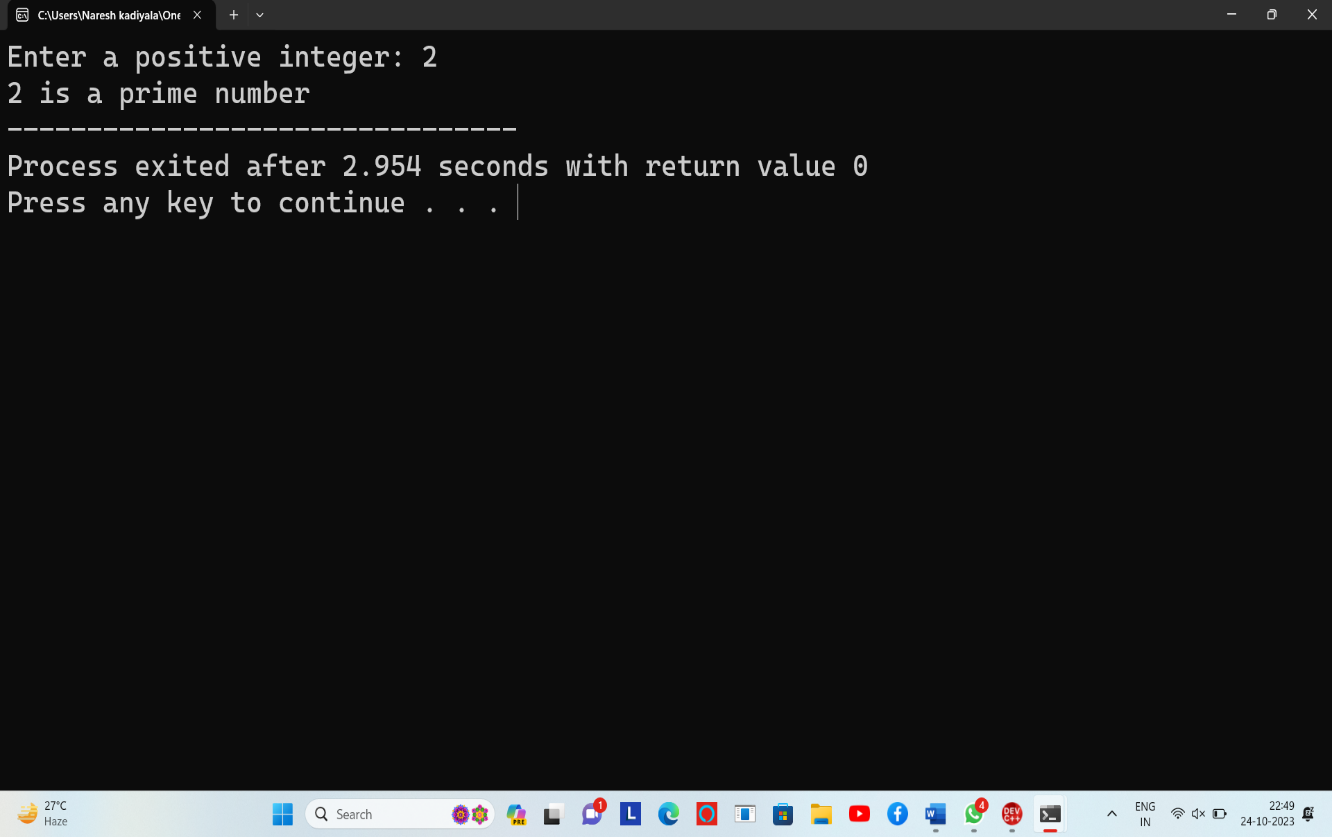
else

cout << n << " is not a prime number";

return 0;

}

**Output:-**

****

**5.)code:-**

#include <iostream>

using namespace std;

class Animal {

public:

virtual void move() {

cout << "This is how an animal moves." << endl;

}

};

class Bird : public Animal {

public:

void move() override {

cout << "A bird flies through the sky." << endl;

}

};

class Fish : public Animal {

public:

void move() override {

cout << "A fish swims in the water." << endl;

}

};

int main() {

Animal\* animal1 = new Animal();

Animal\* animal2 = new Bird();

Animal\* animal3 = new Fish();

cout << "Calling move for Animal: ";

animal1->move();

cout << "Calling move for Bird: ";

animal2->move();

cout << "Calling move for Fish: ";

animal3->move();

delete animal1;

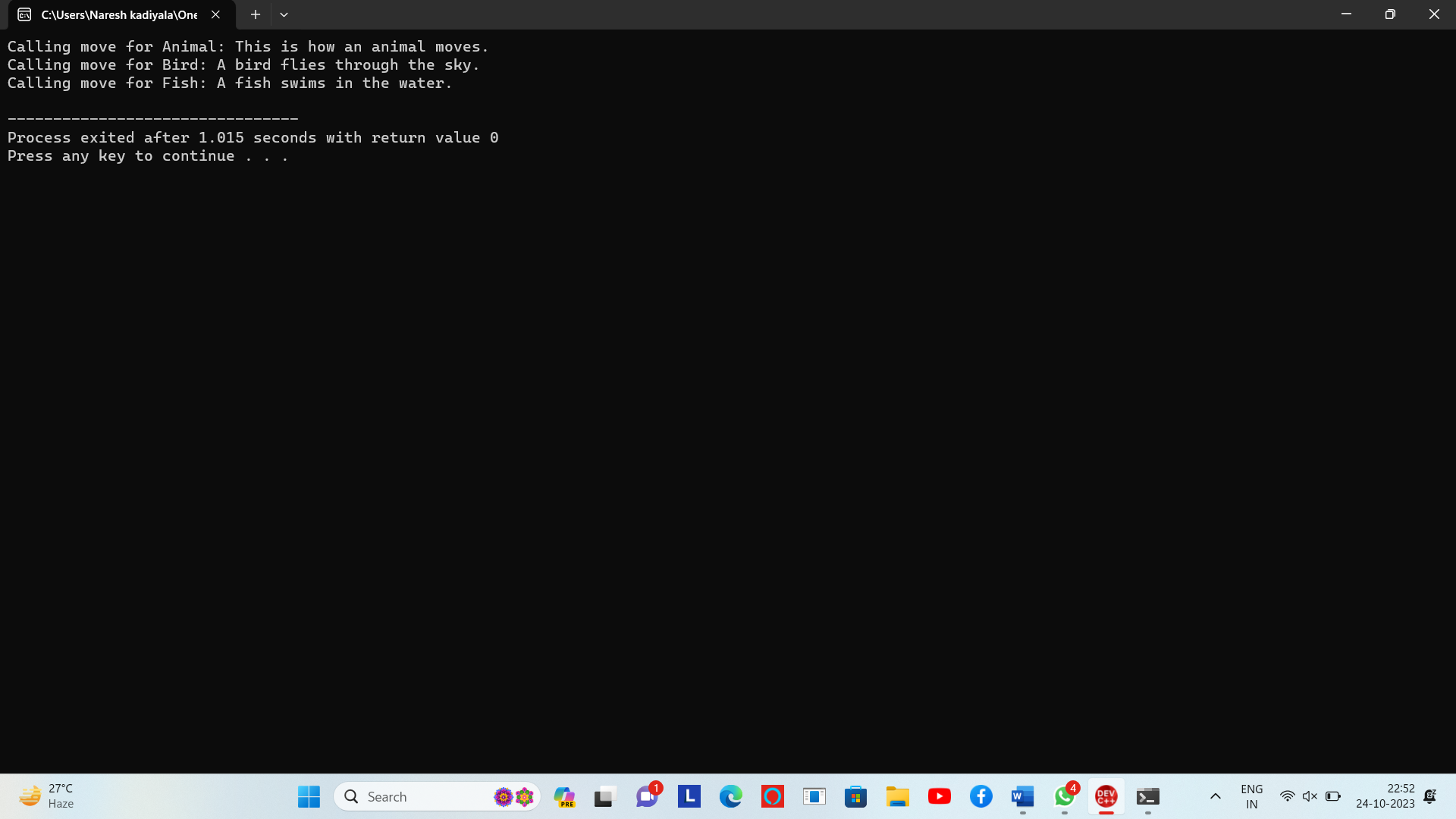
delete animal2;

delete animal3;

return 0;

}

**Output:-**

****

**6.)code:-**

#include <iostream>

#include <string>

using namespace std;

class Employee {

protected:

string name;

int id;

double salary;

public:

Employee(const string& n, int i, double s)

: name(n), id(i), salary(s) {}

void setName(const string& n) {

name = n;

}

string getName() const {

return name;

}

void setId(int i) {

id = i;

}

int getId() const {

return id;

}

void setSalary(double s) {

salary = s;

}

double getSalary() const {

return salary;

}

};

class Manager : public Employee {

private:

string specialty;

public:

Manager(const string& n, int i, double s, const string& spec)

: Employee(n, i, s), specialty(spec) {}

void setSpecialty(const string& spec) {

specialty = spec;

}

std::string getSpecialty() const {

return specialty;

}

};

class Engineer : public Employee {

private:

int hours;

public:

Engineer(const string& n, int i, double s, int h)

: Employee(n, i, s), hours(h) {}

void setHours(int h) {

hours = h;

}

int getHours() const {

return hours;

}

};

int main() {

Manager manager("K.Naresh", 101, 75000.0, "Finance");

Engineer engineer("sai kumar", 102, 30000, 9);

cout << "Manager Details:" << endl;

cout << "Name: " << manager.getName() << endl;

cout << "ID: " << manager.getId() << endl;

cout << "Salary: " << manager.getSalary() << endl;

cout << "Specialty: " << manager.getSpecialty() << endl;

cout << "\nEngineer Details:" << std::endl;

cout << "Name: " << engineer.getName() << endl;

cout << "ID: " << engineer.getId() <<endl;

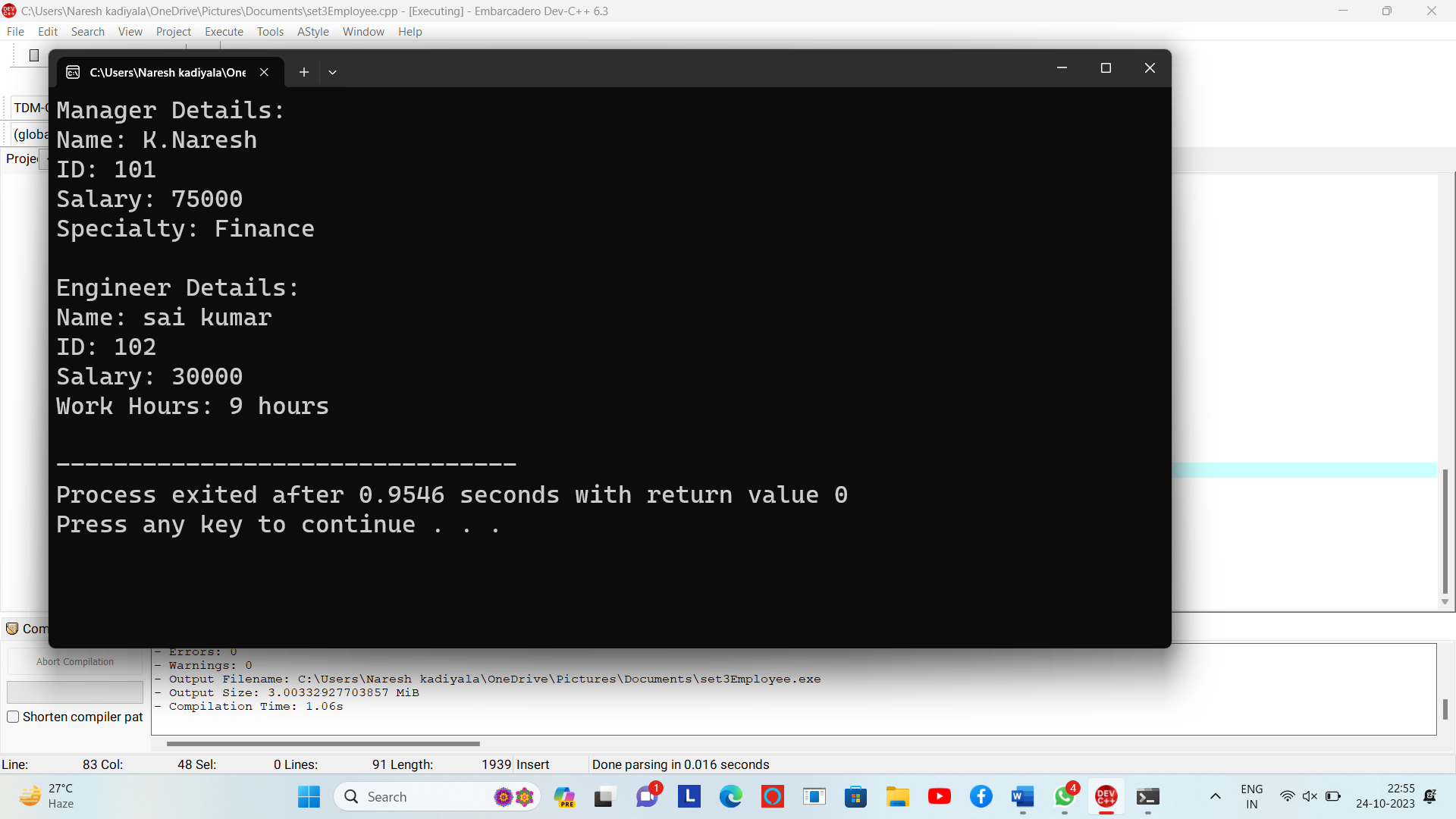
cout << "Salary: " << engineer.getSalary() <<endl;

cout << "Work Hours: " << engineer.getHours() << " hours" <<endl;

return 0;

}

**Output:-**

****

**7.)code:-**

#include<iostream>

using namespace std;

class MyNumber {

private:

double real;

double imaginary;

public:

MyNumber(double r = 0.0, double i = 0.0) {

real = r;

imaginary = i;

}

bool operator==(const MyNumber &other) {

return (real == other.real) && (imaginary == other.imaginary);

}

void show() {

cout << real << " + " << imaginary << "i" << endl;

}

};

int main() {

MyNumber num1(3.0);

MyNumber num2(4.0);

MyNumber num3;

cout << "num1: ";

num1.show();

cout << "num2: ";

num2.show();

if (num1 == num2) {

cout << "num1 and num2 are equal numbers." << endl;

}

else {

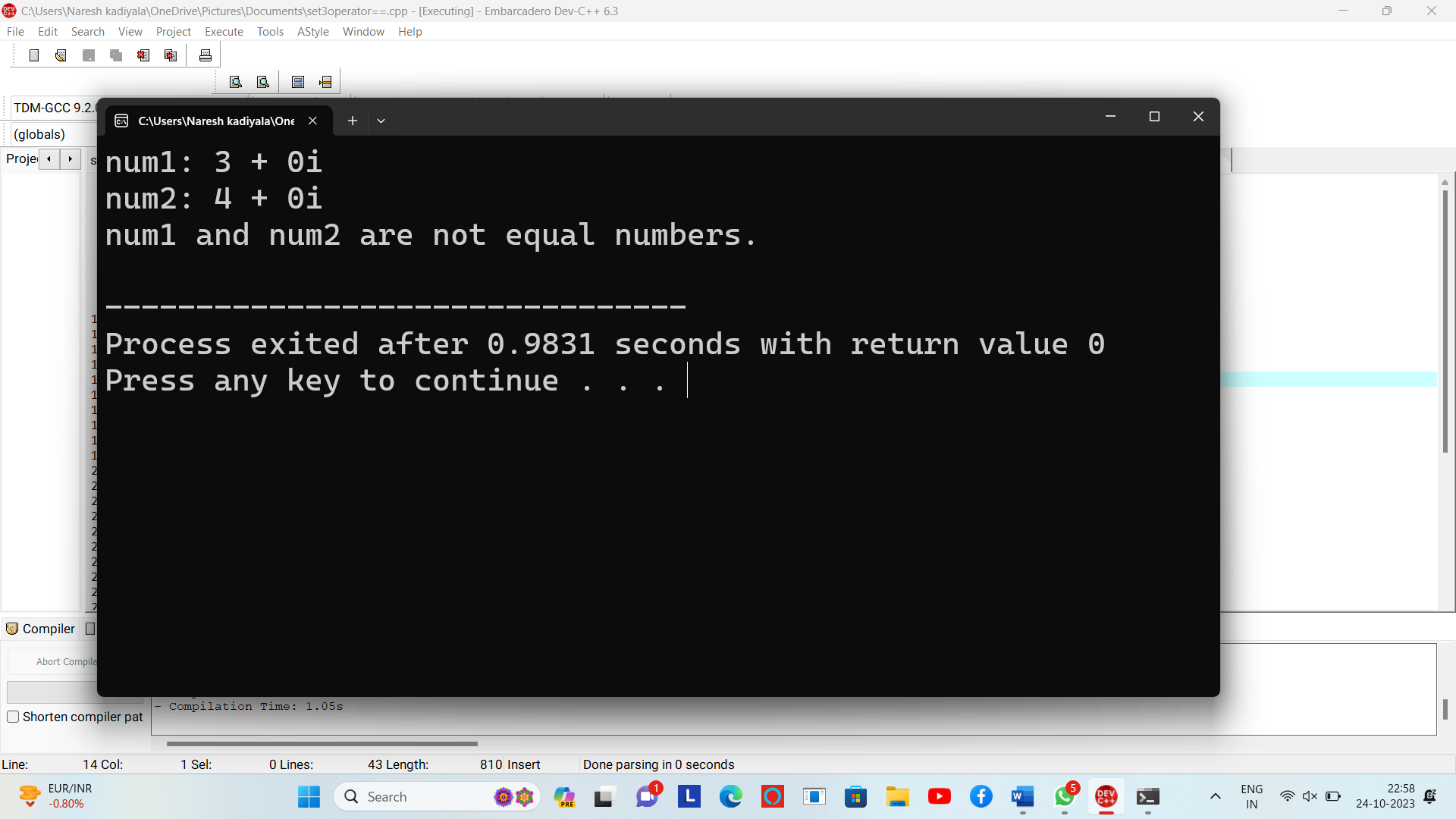
cout << "num1 and num2 are not equal numbers." << endl;

}

return 0;

}

**Output:-**



8.)code:-

#include <iostream>

using namespace std;

class Point {

public:

int x;

int y;

Point(int x, int y) : x(x), y(y) {}

};

int main() {

Point myPoint(10, 20);

Point\* ptr = &myPoint;

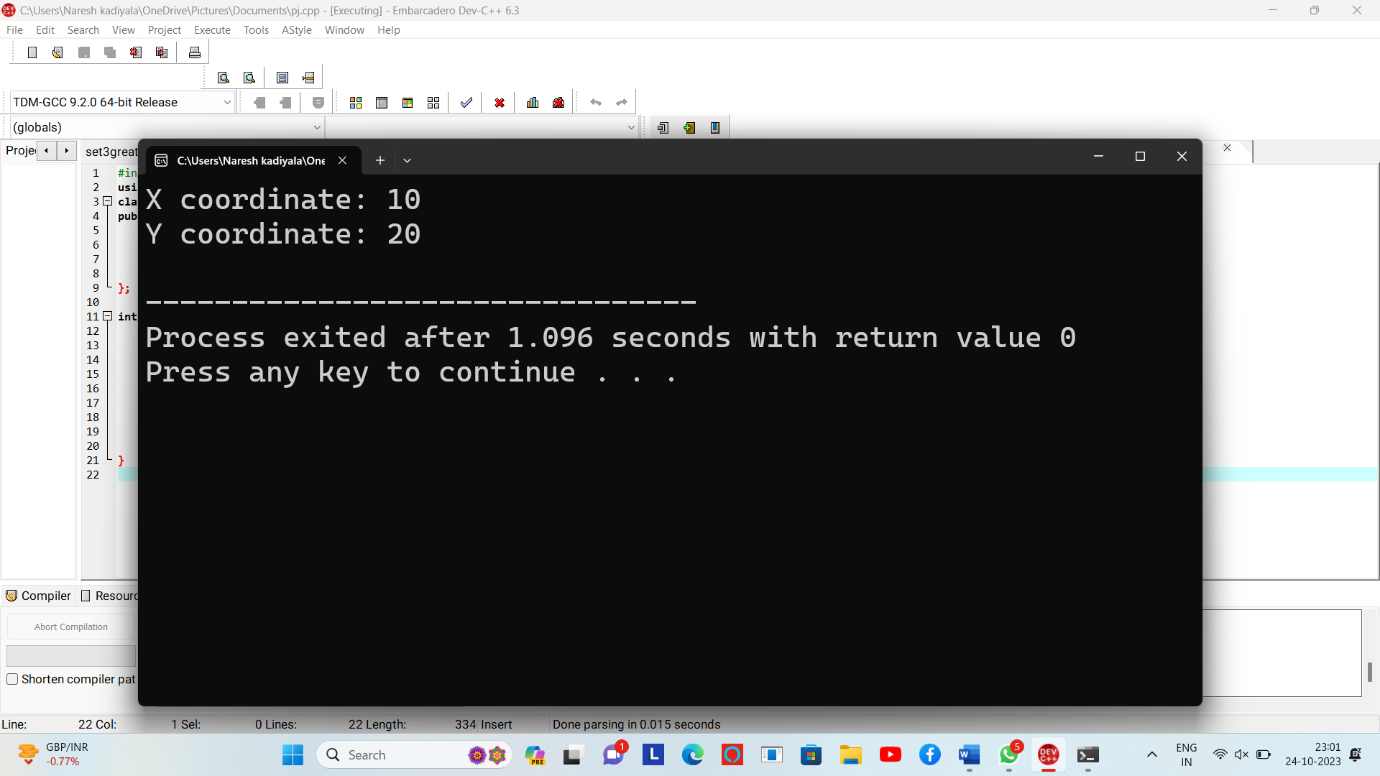
cout << "X coordinate: " << ptr->x << endl;

cout << "Y coordinate: " << ptr->y << endl;

return 0;

}

**Output:-**

****