**PRACTICAL 1**

PART 1

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pracs1sk

{

//inheritance

class A

{

public virtual void display()

{

Console.WriteLine("\nThis is class A");

}

}

class B : A

{

public virtual void display()

{

Console.WriteLine("\nThis is class B");

}

}

class C : B

{

public virtual void display()

{

Console.WriteLine("\nThis is class C");

}

}

internal class Program : C

{

public virtual void display()

{

Console.WriteLine("\nThis is class Program");

}

static void Main(string[] args)

{

A x = new Program();

x.display();

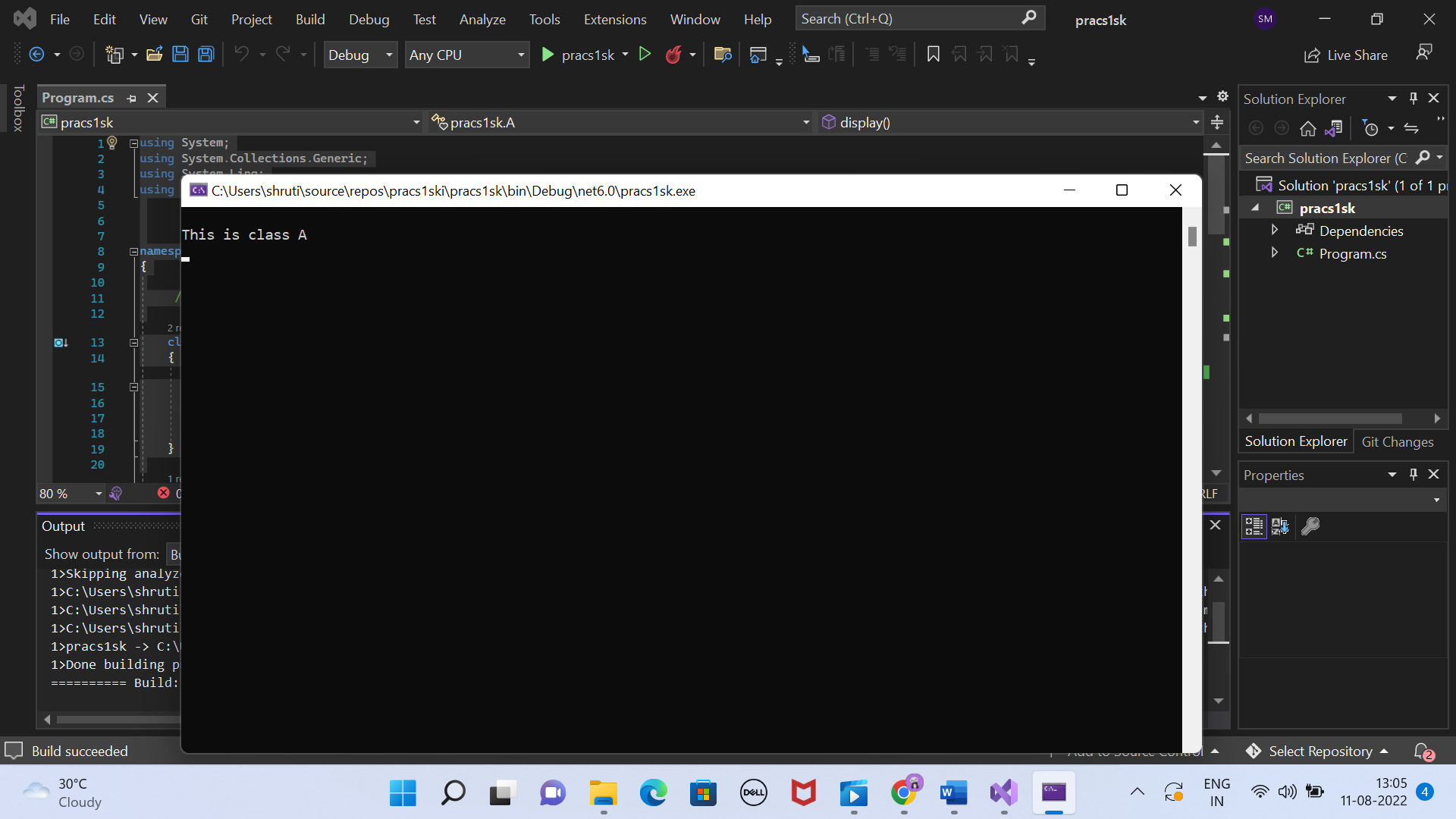
Console.ReadKey();

}

}

}

**OUTPUT:**



PART 2

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pracs1sk

{

//inheritance

class A

{

public virtual void display()

{

Console.WriteLine("\nThis is class A");

}

}

class B : A

{

public override void display()

{

Console.WriteLine("\nThis is class B");

}

}

class C : B

{

public override void display()

{

Console.WriteLine("\nThis is class C");

}

}

internal class Program : C

{

public override void display()

{

Console.WriteLine("\nThis is class Program");

}

static void Main(string[] args)

{

A x = new Program();

x.display();

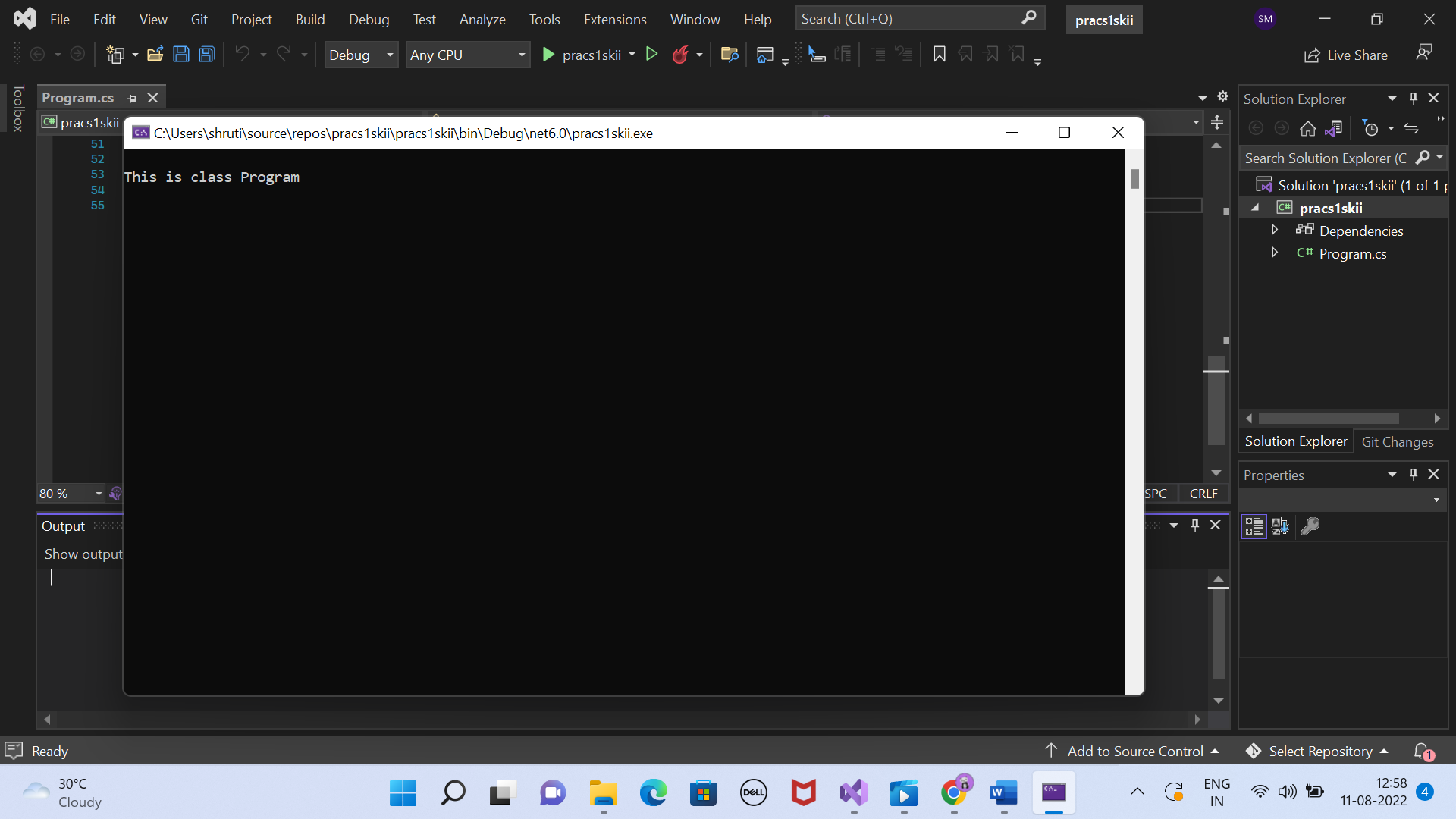
Console.ReadKey();

}

}

}

**OUTPUT:**



PART 3

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pracs1skiii

{

//inheritance

class A

{

public virtual void display()

{

Console.WriteLine("\nThis is class A");

}

}

class B : A

{

public override void display()

{

Console.WriteLine("\nThis is class B");

}

}

class C : B

{

public new virtual void display()

{

Console.WriteLine("\nThis is class C");

}

}

internal class Program : C

{

public override void display()

{

Console.WriteLine("\nThis is class Program");

}

static void Main(string[] args)

{

A x = new Program();

x.display();

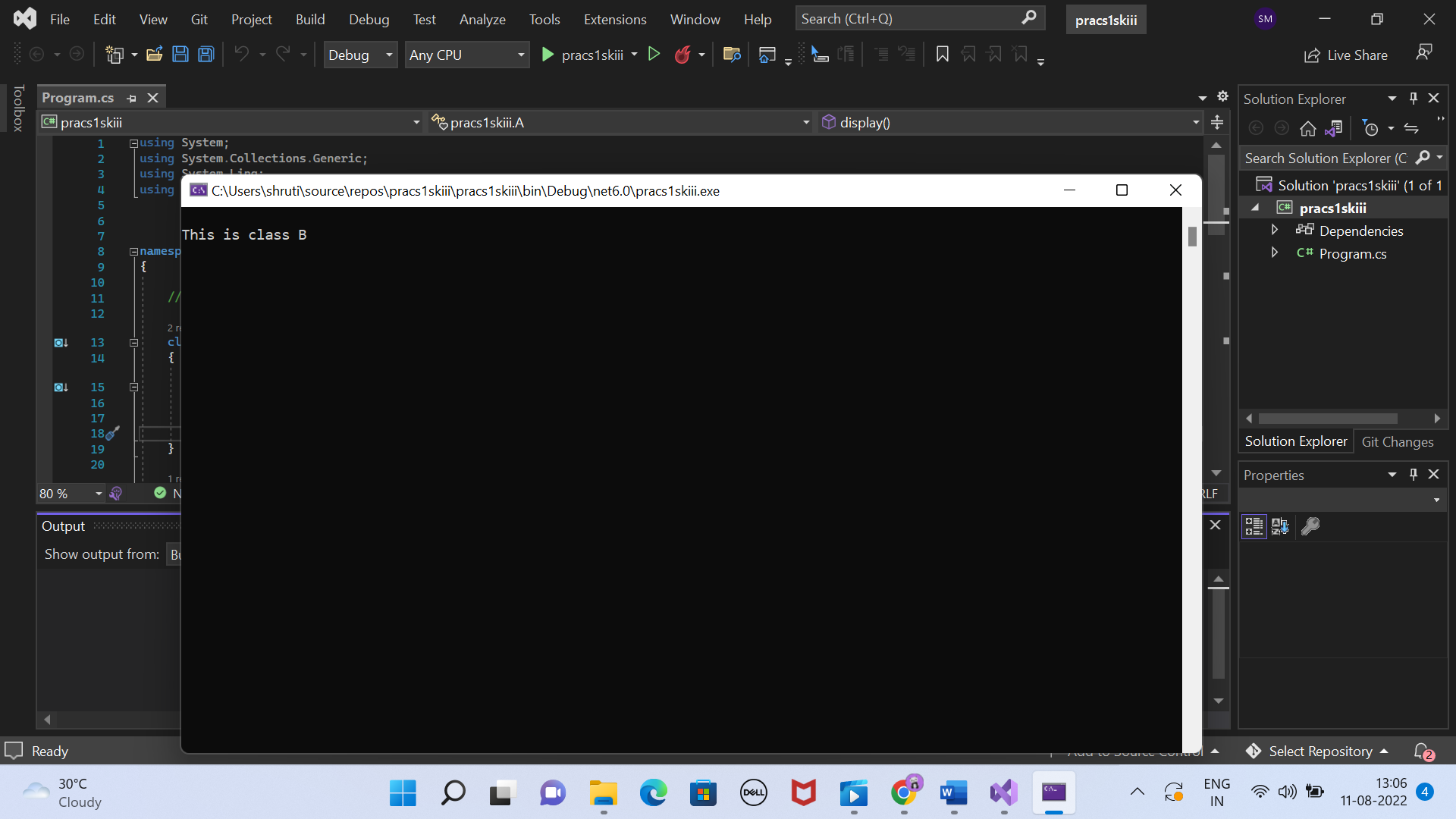
Console.ReadKey();

}

}

}

**OUTPUT:**



PART 4

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pracs1sk

{

//inheritance

class A

{

public virtual void display()

{

Console.WriteLine("\nThis is class A");

}

}

class B : A

{

public override void display()

{

Console.WriteLine("\nThis is class B");

}

}

class C : B

{

public override void display()

{

Console.WriteLine("\nThis is class C");

}

}

internal class Program : C

{

public new virtual void display()

{

Console.WriteLine("\nThis is class Program");

}

static void Main(string[] args)

{

A x = new Program();

x.display();

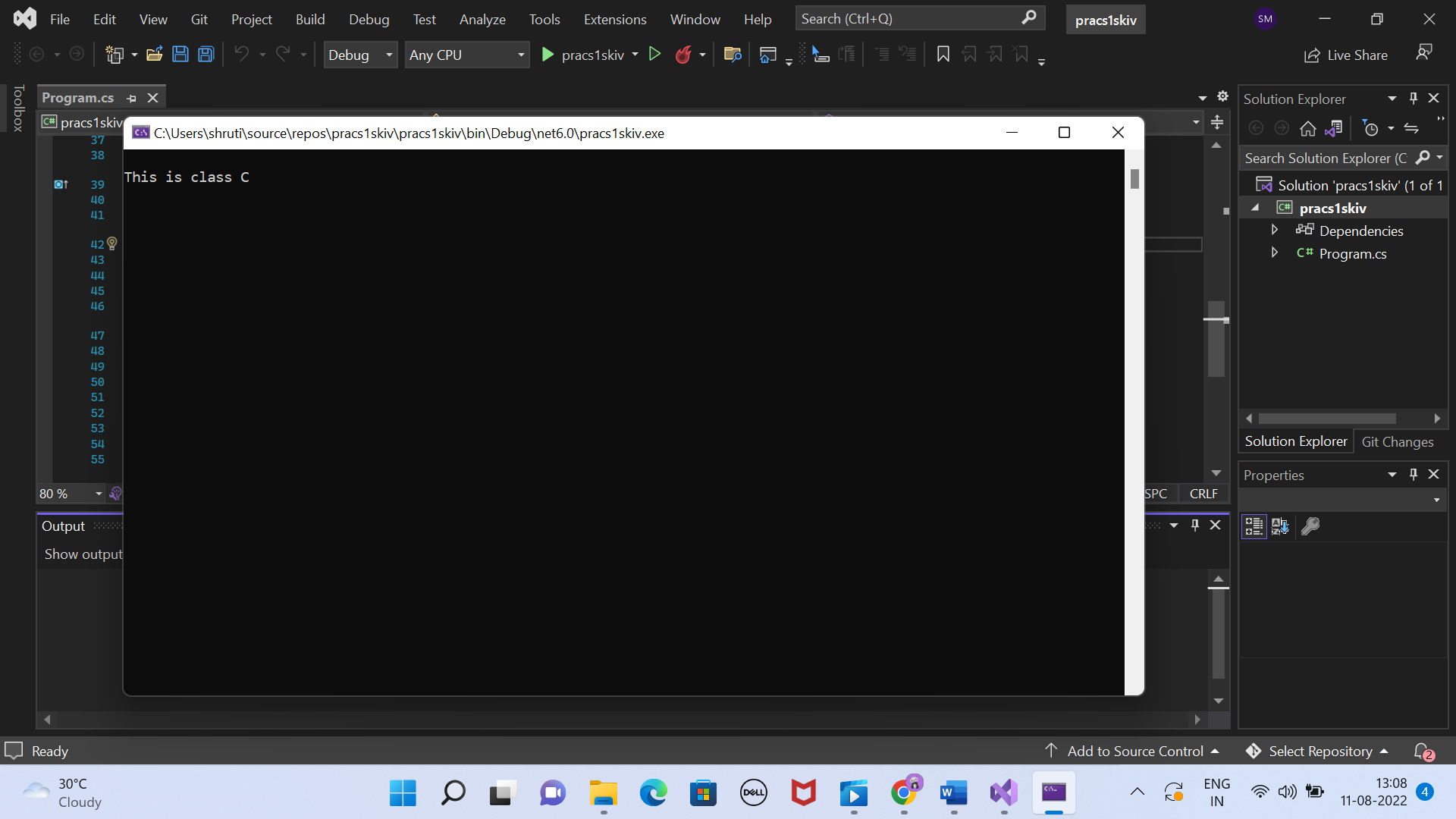
Console.ReadKey();

}

}

}

**OUTPUT:**



**PRACTICAL 2**

PART 1

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace pracs2sk

{

class A

{

public A()

{

Console.WriteLine("This is constructor A");

}

public void display()

{

Console.WriteLine("This is class A");

}

}

class B:A

{

public B()

{

Console.WriteLine("This is constructor B");

}

public void display()

{

Console.WriteLine("This is class B");

}

}

class C:B

{

public C()

{

Console.WriteLine("This is constructor C");

}

public void display()

{

Console.WriteLine("This is class C");

}

}

internal class Program:C

{

Program()

{

Console.WriteLine("This is constructor Program");

}

public void display()

{

Console.WriteLine("This is class Program");

}

static void Main(string[] args)

{

A p = new Program();

p.display();

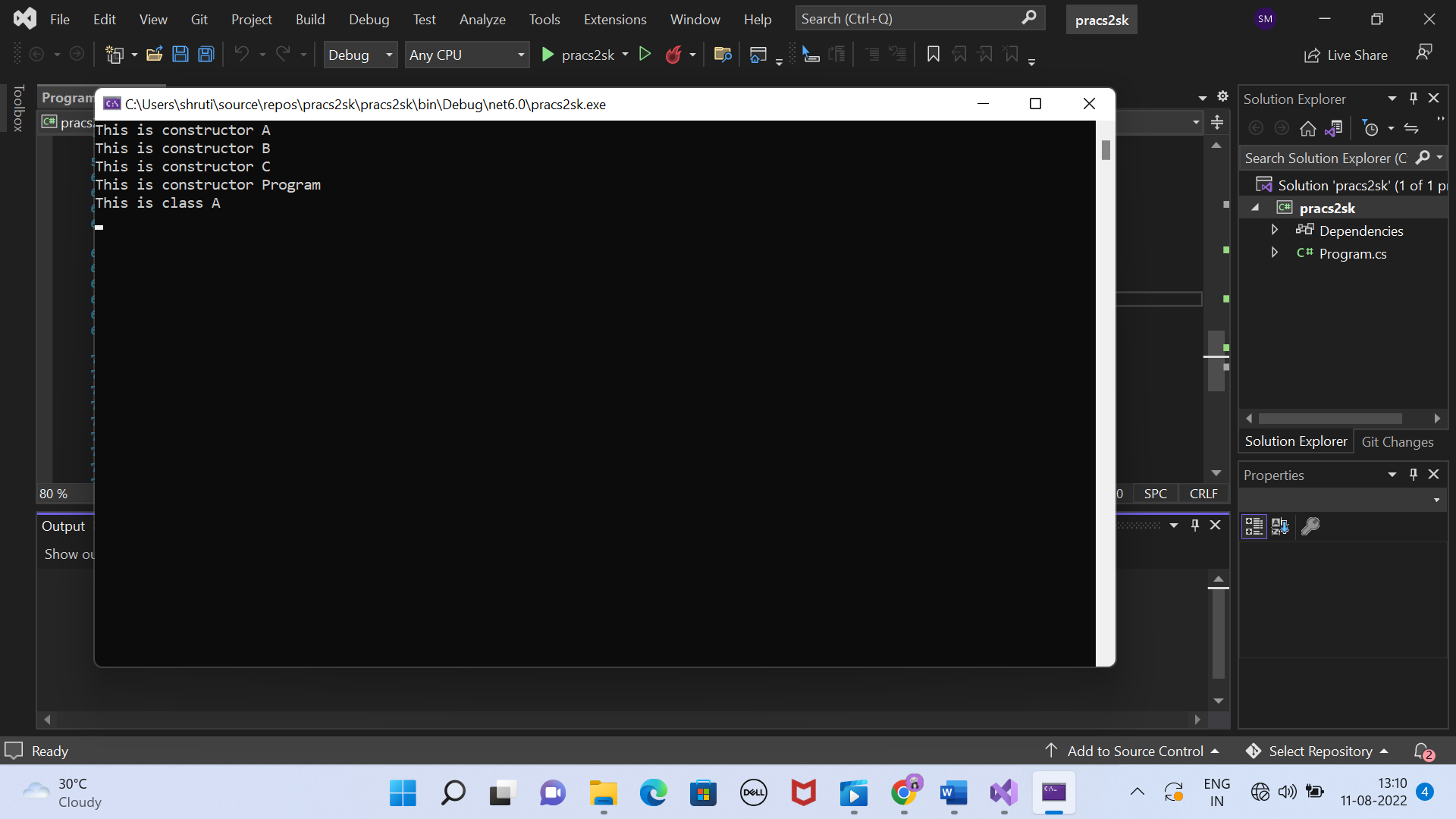
Console.ReadKey();

}

}

}

**OUTPUT:**



PART 2

**INPUT:**

namespace pracs2skii

{

class A

{

public A()

{

Console.WriteLine("This is constructor A");

}

public virtual void display()

{

Console.WriteLine("This is class A");

}

}

class B:A

{

public B()

{

Console.WriteLine("This is constructor B");

}

public override void display()

{

Console.WriteLine("This is class B");

}

}

class C:B

{

public C()

{

Console.WriteLine("This is constructor C");

}

public virtual void display()

{

Console.WriteLine("This is class C");

}

}

internal class Program : C

{

public Program()

{

Console.WriteLine("This is constructor Program");

}

public virtual void display()

{

Console.WriteLine("This is class Program");

}

static void Main(string[] args)

{

A x = new Program();

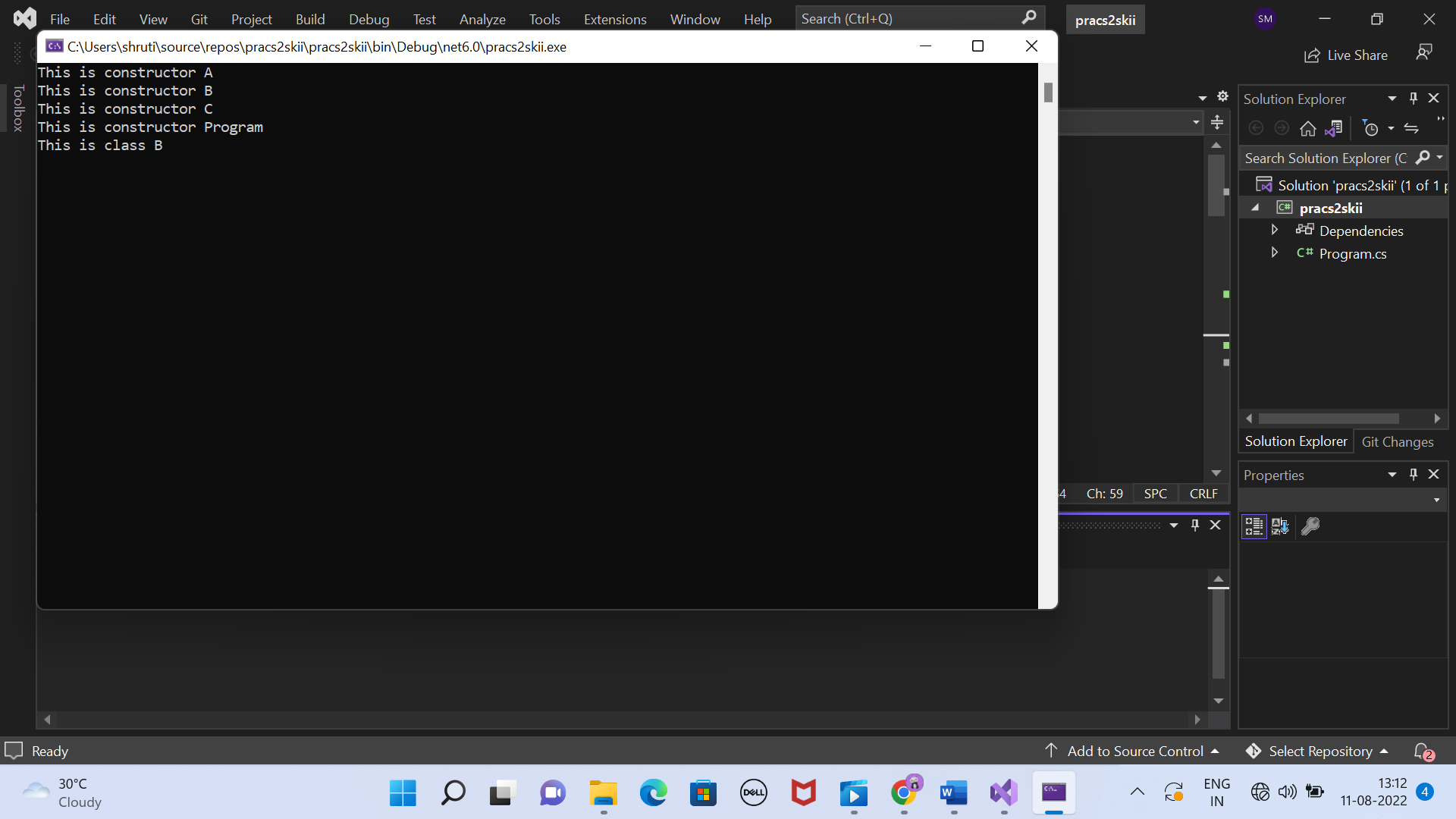
x.display();

Console.ReadKey();

}

}

**OUTPUT:**



PART 3

**INPUT:**

namespace pracs2skiii

{

class A

{

public A()

{

Console.WriteLine("This is constructor A");

}

public virtual void display()

{

Console.WriteLine("This is class A");

}

}

class B : A

{

public B()

{

Console.WriteLine("This is constructor B");

}

public override void display()

{

Console.WriteLine("This is class B");

}

}

class C : A

{

public C()

{

Console.WriteLine("This is constructor C");

}

public override void display()

{

Console.WriteLine("This is class C");

}

}

internal class Program : C

{

public Program()

{

Console.WriteLine("This is constructor Program");

}

public new virtual void display()

{

Console.WriteLine("This is class Program");

}

static void Main(string[] args)

{

A x = new Program();

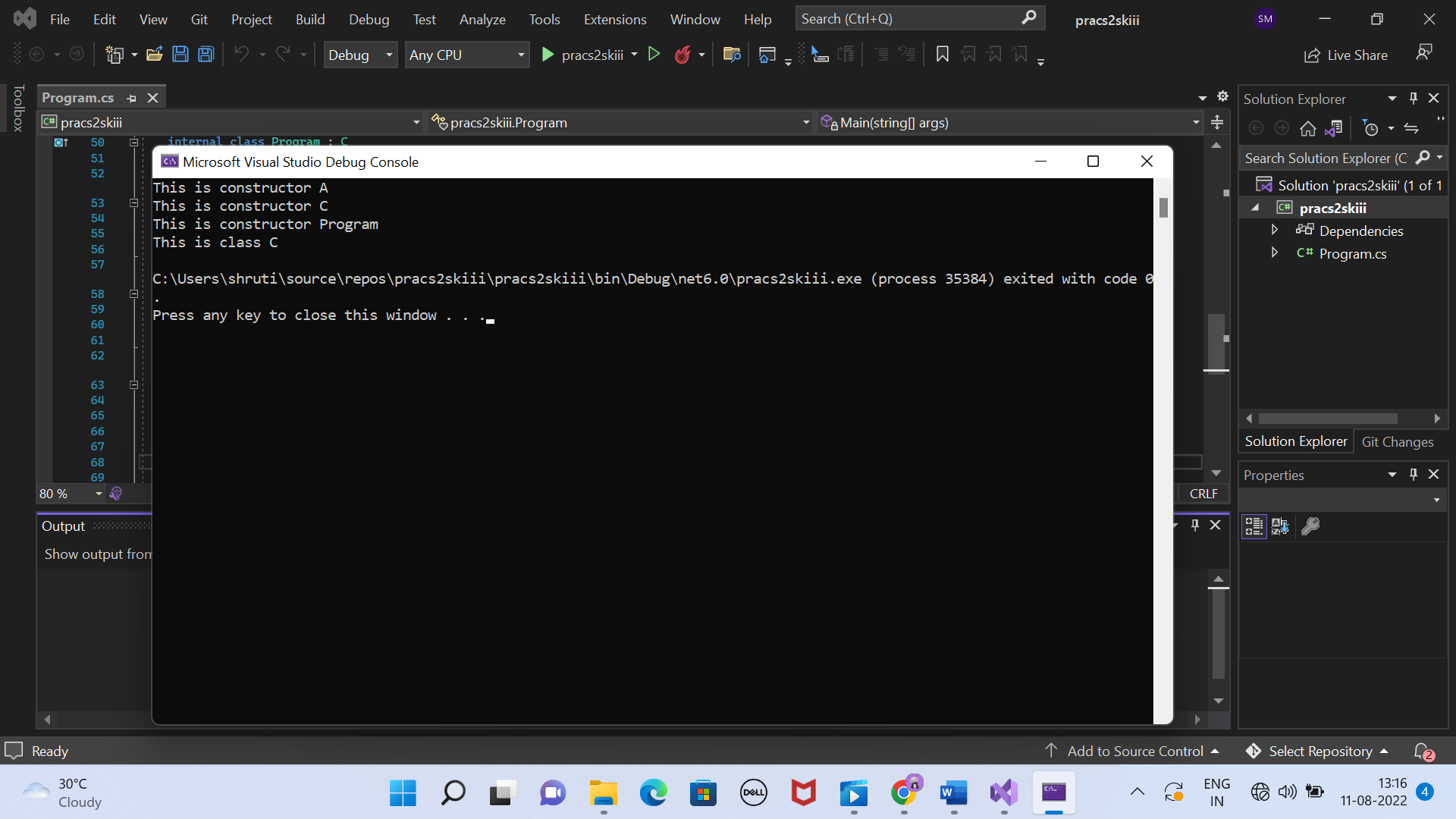
x.display();

}

}

}

**OUTPUT:**



PART 4

**INPUT:**

namespace pracs2skiv

{

class A

{

public A()

{

Console.WriteLine("This is constructor A");

}

public virtual void display()

{

Console.WriteLine("This is class A");

}

}

class B : A

{

public B()

{

Console.WriteLine("This is constructor B");

}

public override void display()

{

Console.WriteLine("This is class B");

}

}

class C : A

{

public C()

{

Console.WriteLine("This is constructor C");

}

public override void display()

{

Console.WriteLine("This is class C");

}

}

internal class Program : C

{

public Program()

{

Console.WriteLine("This is constructor Program");

}

public override void display()

{

Console.WriteLine("This is class Program");

}

static void Main(string[] args)

{

A x = new Program();

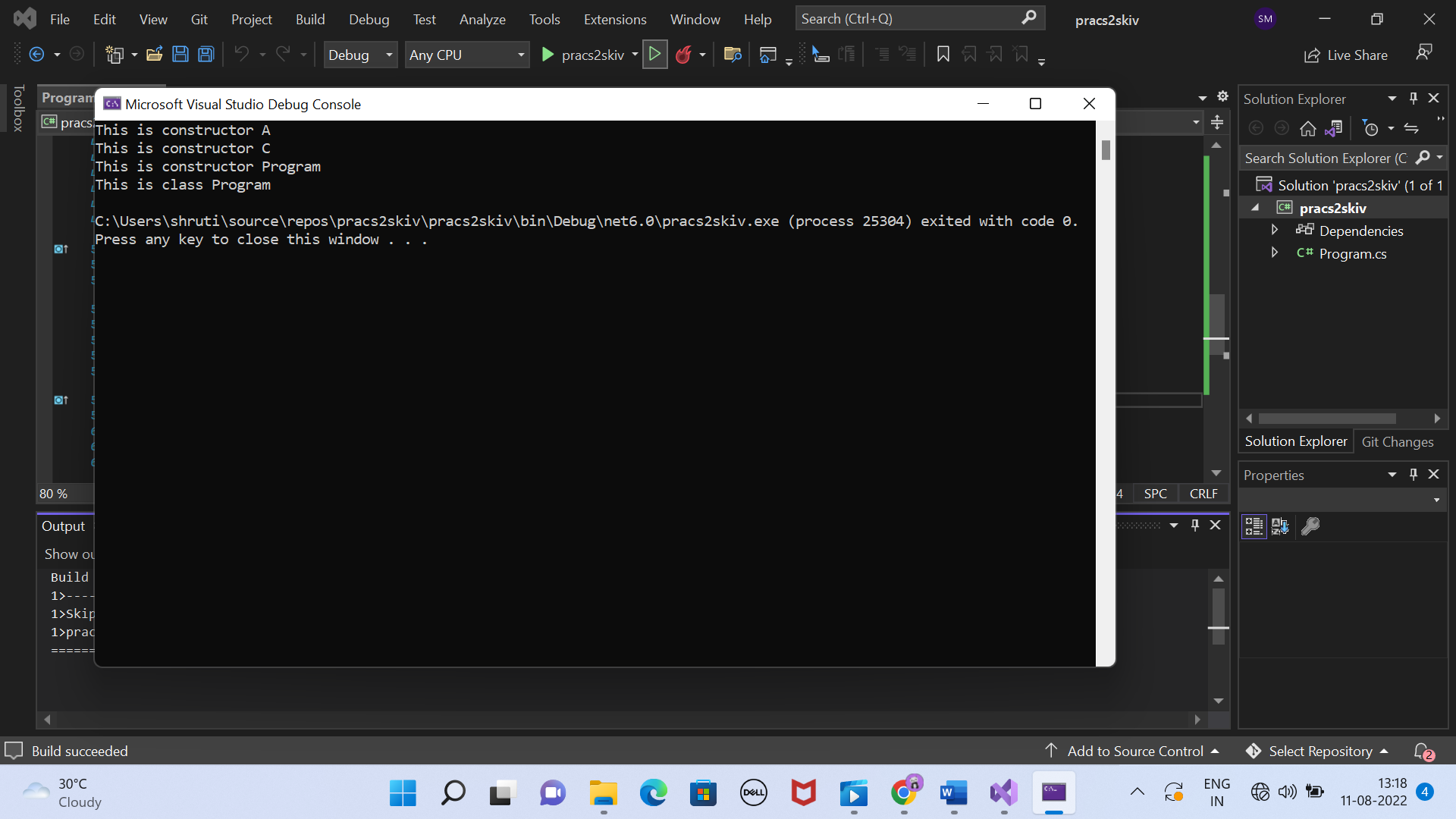
x.display();

}

}

}

**OUTPUT:**



**PRACTICAL 3**

PART (all parts included in one)

**INPUT:**

namespace pracs2sk

{

internal class Program

{

struct Student

{

public String name;

public int age;

public String course;

public int uid;

public int marks;

public Student(String name, int age, String course, int uid, int marks)

{

this.name = name;

this.age = age;

this.course = course;

this.uid = uid;

this.marks = marks;

}

}

static void Main(string[] args)

{

int a;

Console.Write("number students data you want to input: ");

// age = int.Parse(Console.ReadLine());

a = Convert.ToInt32(Console.ReadLine());

Student[] s = new Student[a];

String name;

int age;

String course;

int uid;

int marks;

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

{

Console.Write(" \n");

Console.WriteLine("Student " + (i+1));

Console.Write("Name: ");

name = Console.ReadLine();

Console.Write("Age: ");

// age = int.Parse(Console.ReadLine());

age = Convert.ToInt32(Console.ReadLine());

Console.Write("Marks: ");

// marks = int.Parse(Console.ReadLine());

marks = Convert.ToInt32(Console.ReadLine());

Console.Write("Course: ");

course = Console.ReadLine();

Console.Write("uid: ");

// age = int.Parse(Console.ReadLine());

uid = Convert.ToInt32(Console.ReadLine());

s[i] = new Student(name, age, course, uid, marks);

}

Console.Write(" \n");

Console.WriteLine("Record stored");

Console.Write(" \n");

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

{

Console.WriteLine((i+1) + ") "+ s[i].name + " " + s[i].age + " "+ s[i].course + " " + s[i].uid + " " + s[i].marks);

}

Console.Write(" \n");

int uidcheck;

Console.Write("uid:");

// age = int.Parse(Console.ReadLine());

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

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

{

if (s[i].uid == uidcheck)

{

Console.WriteLine((i + 1) + " " + s[i].name + " " + s[i].age + " " + s[i].course + " " + s[i].uid + " " + s[i].marks);

}

}

Console.Write(" \n");

Console.Write("Course Name :");

String identity = Console.ReadLine();

int sum = 0;

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

{

if (s[i].course == identity)

{

Console.WriteLine("Student name : " + s[i].name + " Marks : " + s[i].marks);

sum += s[i].marks;

}

}

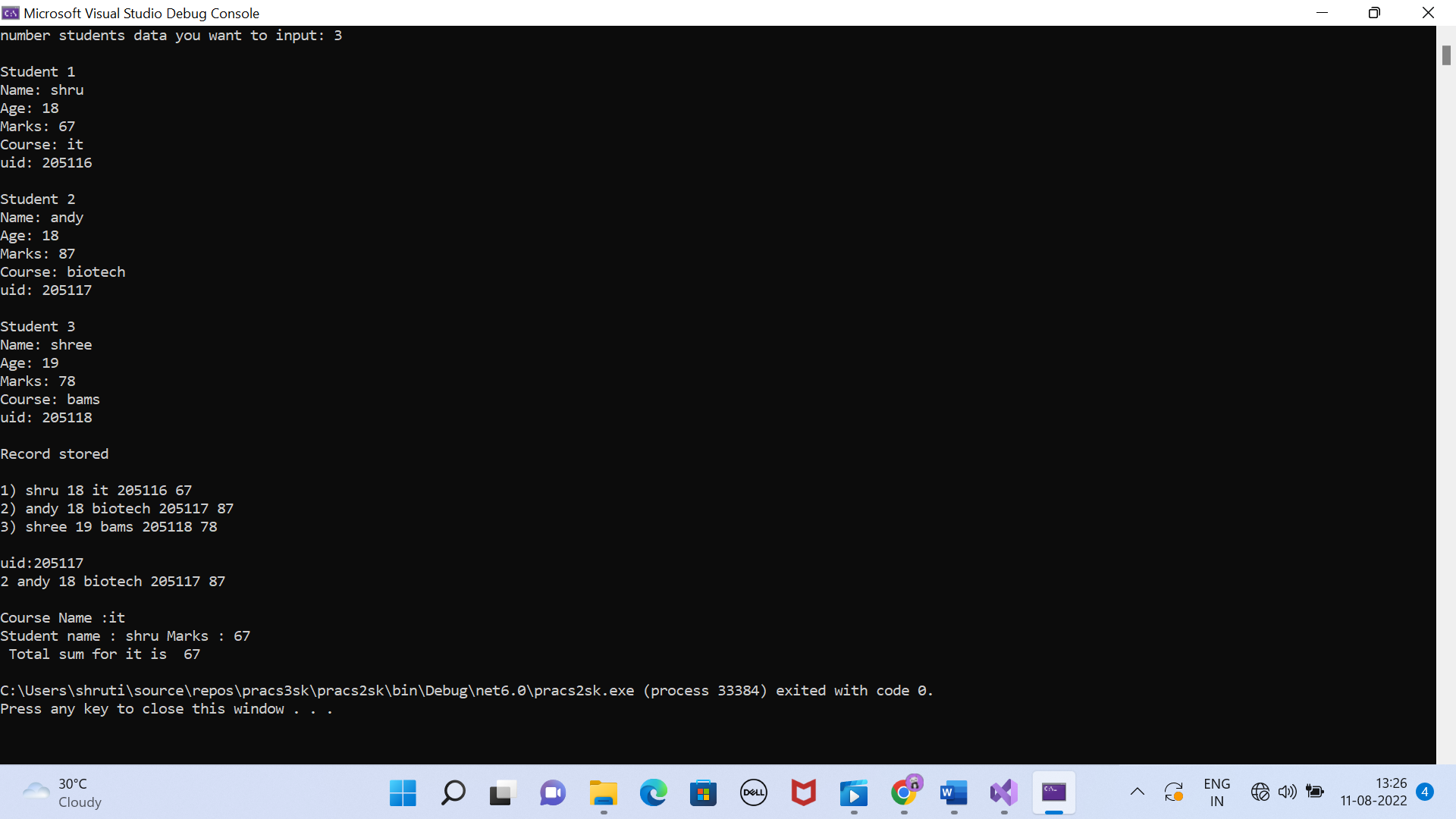
Console.WriteLine(" Total sum for " + identity + " is " + sum);

}

}

}

**OUTPUT:**



**PRACTICAL 4**

PART 1

**INPUT:**

namespace pracs4sk

{

internal class Program

{

static void Main(string[] args)

{

int[,] n1 = new int[3, 3];

int[,] n2 = new int[3, 3];

int[,] n3 = new int[3, 3];

//making of n1

for (int i = 0; i < 3; i++)

{

Console.WriteLine("enter values for row " + (i + 1));

for (int j = 0; j < 3; j++)

{

n1[i,j] = int.Parse(Console.ReadLine());

}

}

//display n1

Console.WriteLine("\n\nMatrix 1 (m1):");

for (int i = 0; i < 3; i++)

{

Console.WriteLine("\n");

for (int j = 0; j < 3; j++)

{

Console.Write(" \t"+n1[i,j]);

}

}

Console.WriteLine("\n");

//making of n2

for (int i = 0; i < 3; i++)

{

Console.WriteLine("enter values for row " + (i + 1));

for (int j = 0; j < 3; j++)

{

n2[i, j] = int.Parse(Console.ReadLine());

}

}

//display n2

Console.WriteLine("\n\nMatrix 2 (m2):");

for (int i = 0; i < 3; i++)

{

Console.WriteLine("\n");

for (int j = 0; j < 3; j++)

{

Console.Write("\t " + n2[i, j]);

}

}

Console.WriteLine("\n\nvalue of multiplication of Matrix 1 and Matrix 2(m1xm2):");

//multiplication

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

for(int k = 0; k < 3; k++)

{

n3[i, j] = n3[i, j] + n1[i,k] \* n2[k,j];

}

}

}

//display n3

for (int i = 0; i < 3; i++)

{

Console.WriteLine("\n");

for (int j = 0; j < 3; j++)

{

Console.Write("\t " + n3[i, j]);

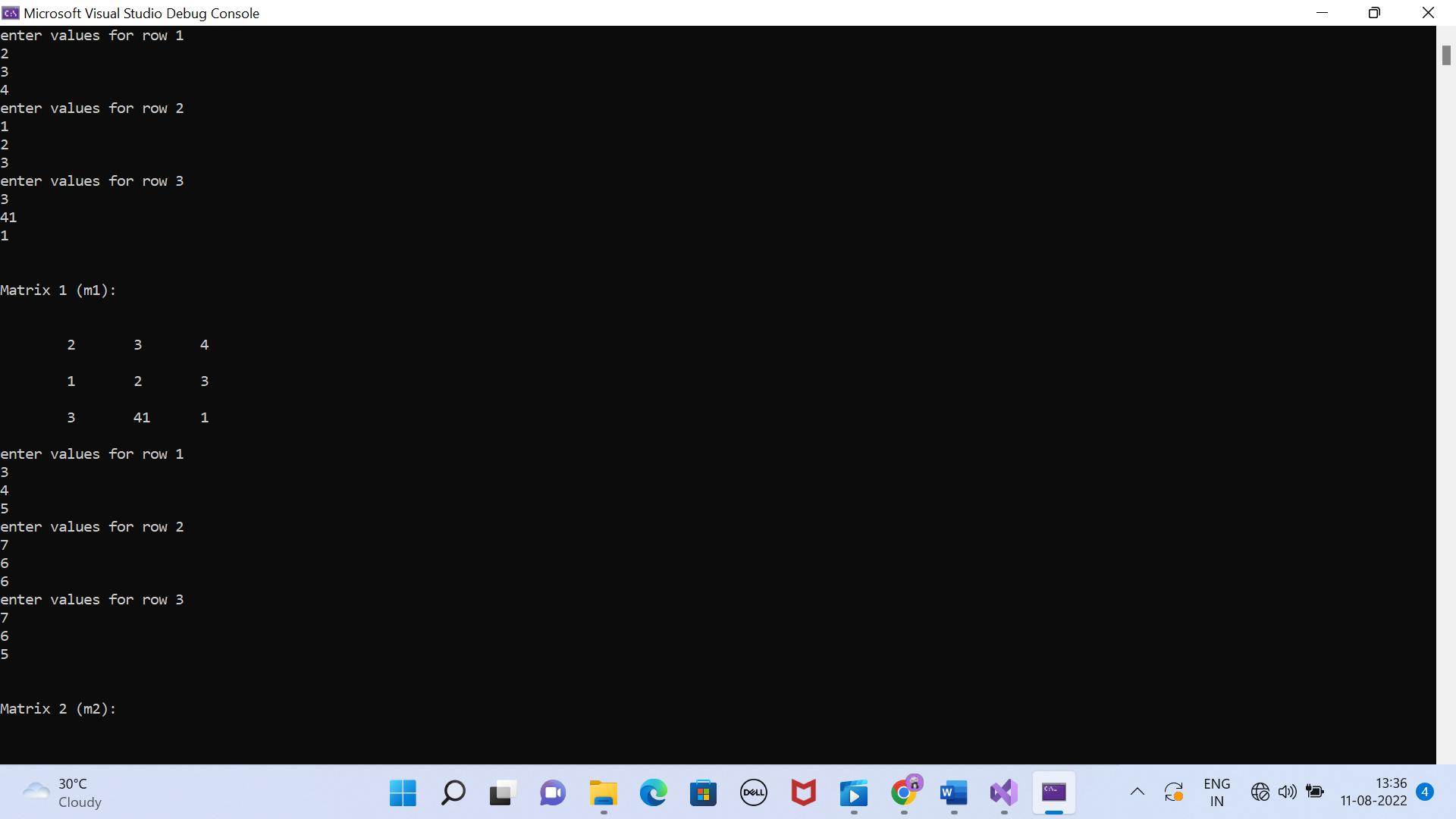
}

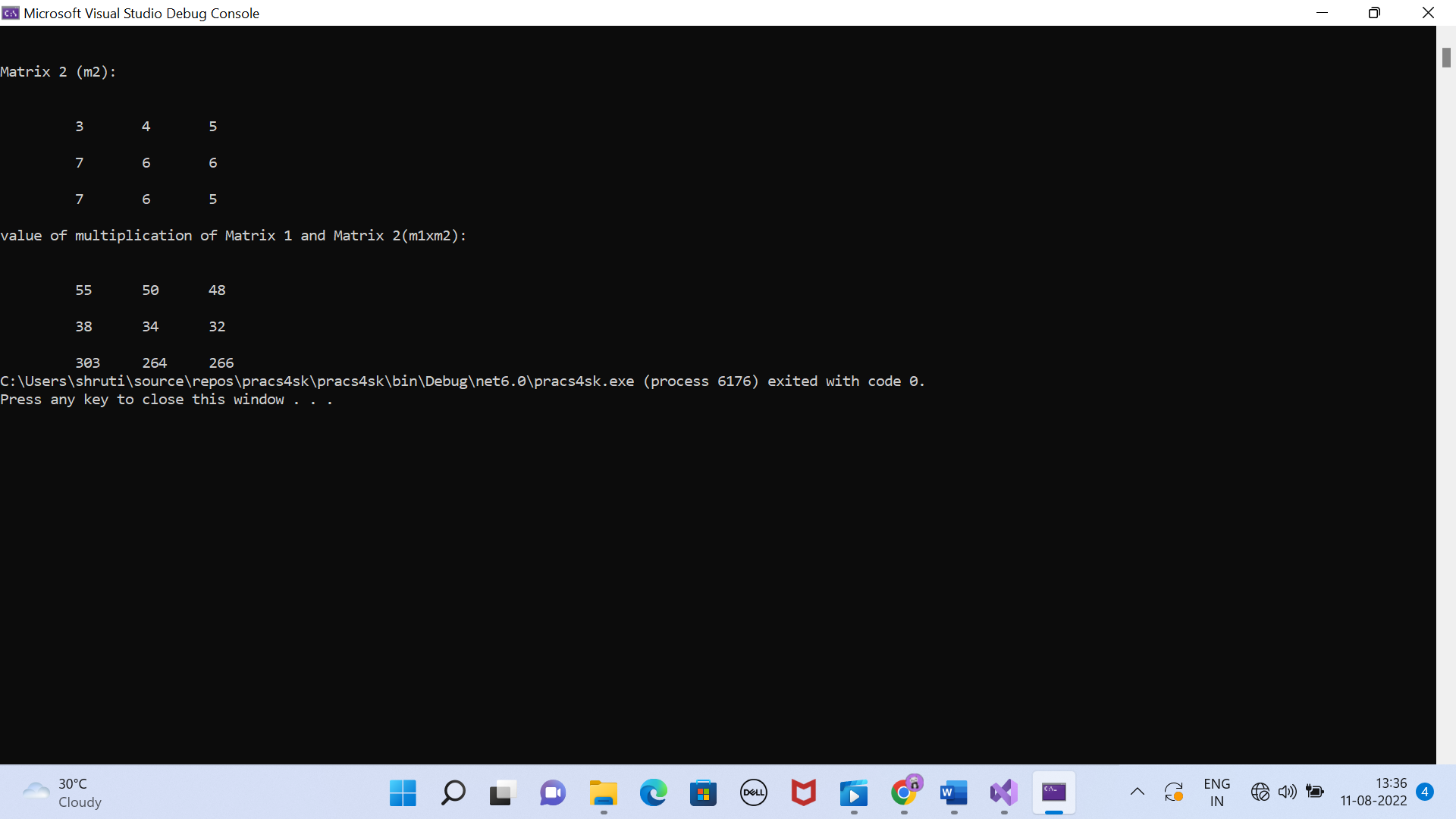
}

}

}

**OUTPUT:**





PART 1

**INPUT:**

**OUTPUT:**

**PRACTICAL 5**

PART a

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Xml.Linq;

using System.Threading.Tasks;

namespace pracs5sk

{

internal class Program

{

static void Main(string[] args)

{

XDocument pd = XDocument.Load("C:\\Users\\shruti\\source\\repos\\pracs5sk\\pracs5sk\\XMLFile1.xml");

var product = pd.Elements();

Console.WriteLine("=== display all product===");

foreach(var item in product)

{

Console.WriteLine(item.ToString());

}

Console.WriteLine();

//part a

Console.WriteLine("part a: more than 55000");

var result = product.Elements("Product") //important line

.Where(x => (int)x.Element("price") > 55000) // specifing data type is necessary

.Select(x => //generic

new //to access more than one column

{

Name = x.Element("pname").Value,

Price = x.Element("price").Value,

Dom = x.Element("dom").Value,

}

);

foreach (var r in result)

{

Console.WriteLine(r.ToString());

}

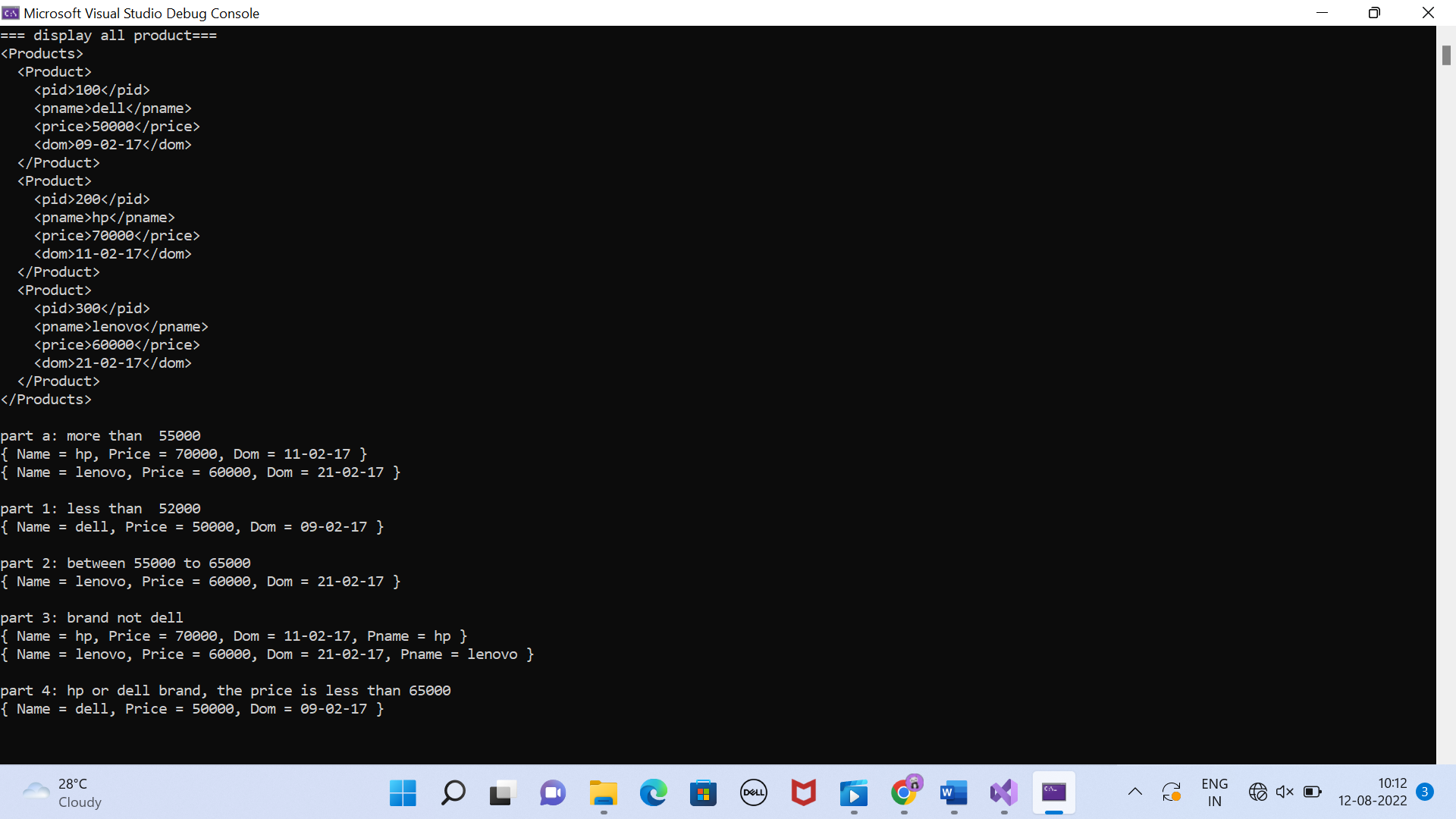
Console.WriteLine();

}

}

}

**OUTPUT:**



PART 1

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Xml.Linq;

using System.Threading.Tasks;

namespace pracs5sk

{

internal class Program

{

static void Main(string[] args)

{

XDocument pd = XDocument.Load("C:\\Users\\shruti\\source\\repos\\pracs5sk\\pracs5sk\\XMLFile1.xml");

var product = pd.Elements();

//part 1

Console.WriteLine("part 1: less than 52000");

var result1 = product.Elements("Product") //important line

.Where(x => (int)x.Element("price") < 52000) // specifing data type is necessary

.Select(x => //generic

new //to access more than one column

{

Name = x.Element("pname").Value,

Price = x.Element("price").Value,

Dom = x.Element("dom").Value,

}

);

foreach (var r in result1)

{

Console.WriteLine(r.ToString());

}

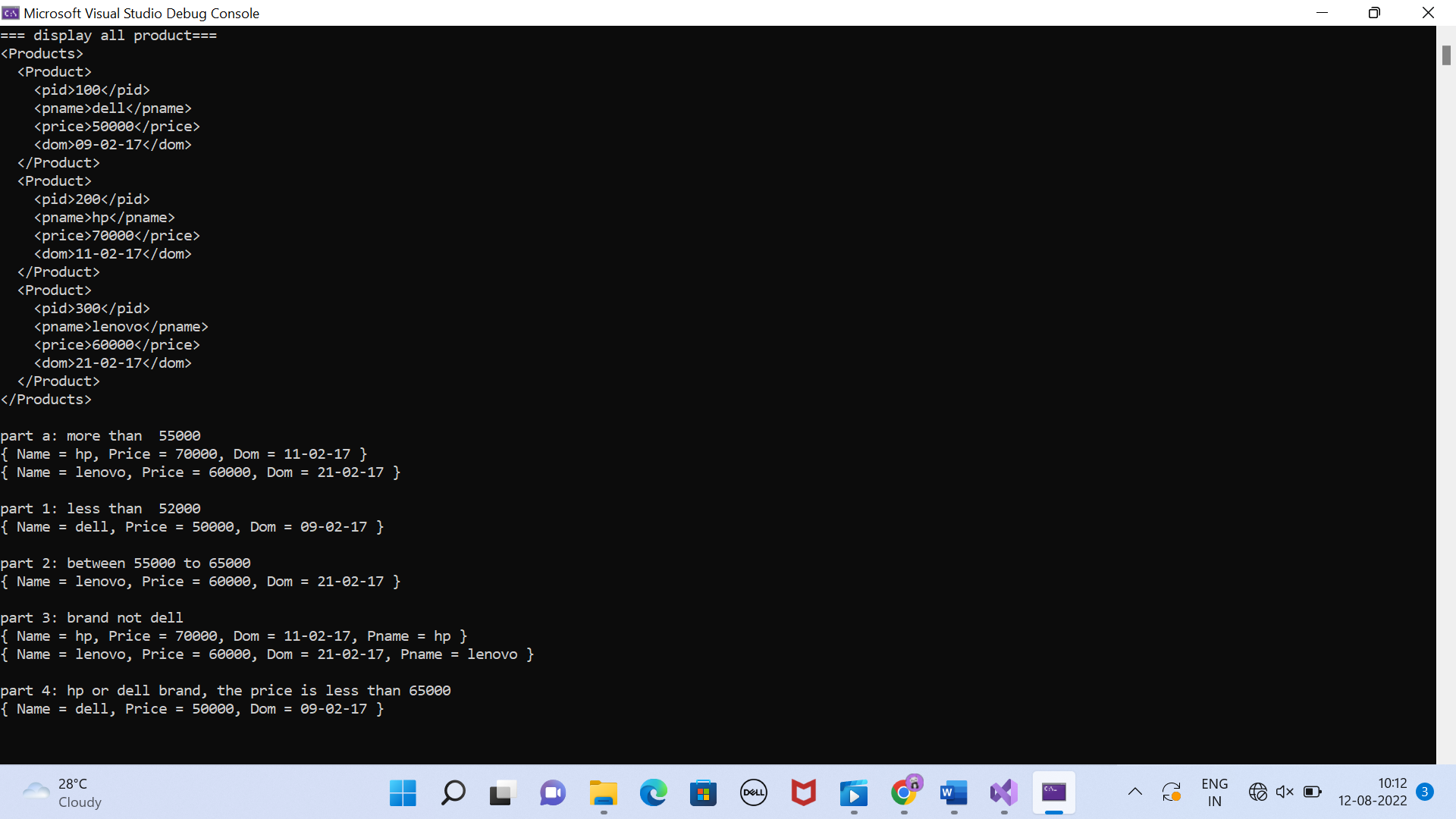
Console.WriteLine();

}

}

}

**OUTPUT:**



PART 2

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Xml.Linq;

using System.Threading.Tasks;

namespace pracs5sk

{

internal class Program

{

static void Main(string[] args)

{

XDocument pd = XDocument.Load("C:\\Users\\shruti\\source\\repos\\pracs5sk\\pracs5sk\\XMLFile1.xml");

var product = pd.Elements();

Console.WriteLine("part 2: between 55000 to 65000");

var result2 = product.Elements("Product") //important line

.Where(x => (int)x.Element("price") > 55000 && (int)x.Element("price") < 65000) // specifing data type is necessary

.Select(x => //generic

new //to access more than one column

{

Name = x.Element("pname").Value,

Price = x.Element("price").Value,

Dom = x.Element("dom").Value,

}

);

foreach (var r in result2)

{

Console.WriteLine(r.ToString());

}

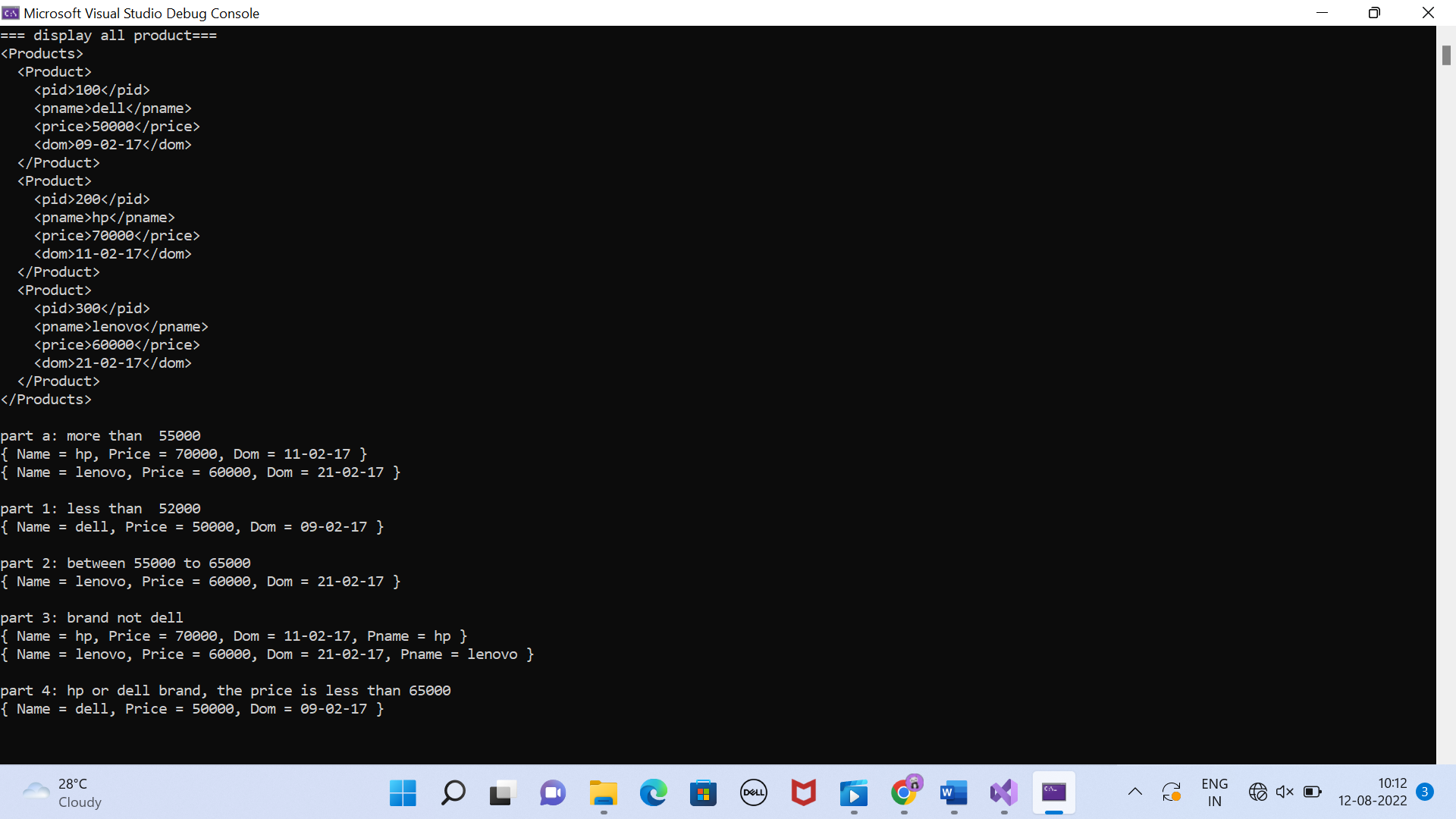
Console.WriteLine();

}

}

}

**OUTPUT:**



PART 3

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Xml.Linq;

using System.Threading.Tasks;

namespace pracs5sk

{

internal class Program

{

static void Main(string[] args)

{

XDocument pd = XDocument.Load("C:\\Users\\shruti\\source\\repos\\pracs5sk\\pracs5sk\\XMLFile1.xml");

var product = pd.Elements();

Console.WriteLine("part 3: brand not dell");

var result3 = product.Elements("Product") //important line

.Where(x => x.Element("pname").Value != "dell") // specifing data type is necessary

.Select(x => //generic

new //to access more than one column

{

Name = x.Element("pname").Value,

Price = x.Element("price").Value,

Dom = x.Element("dom").Value,

Pname = x.Element("pname").Value,

}

);

foreach (var r in result3)

{

Console.WriteLine(r.ToString());

}

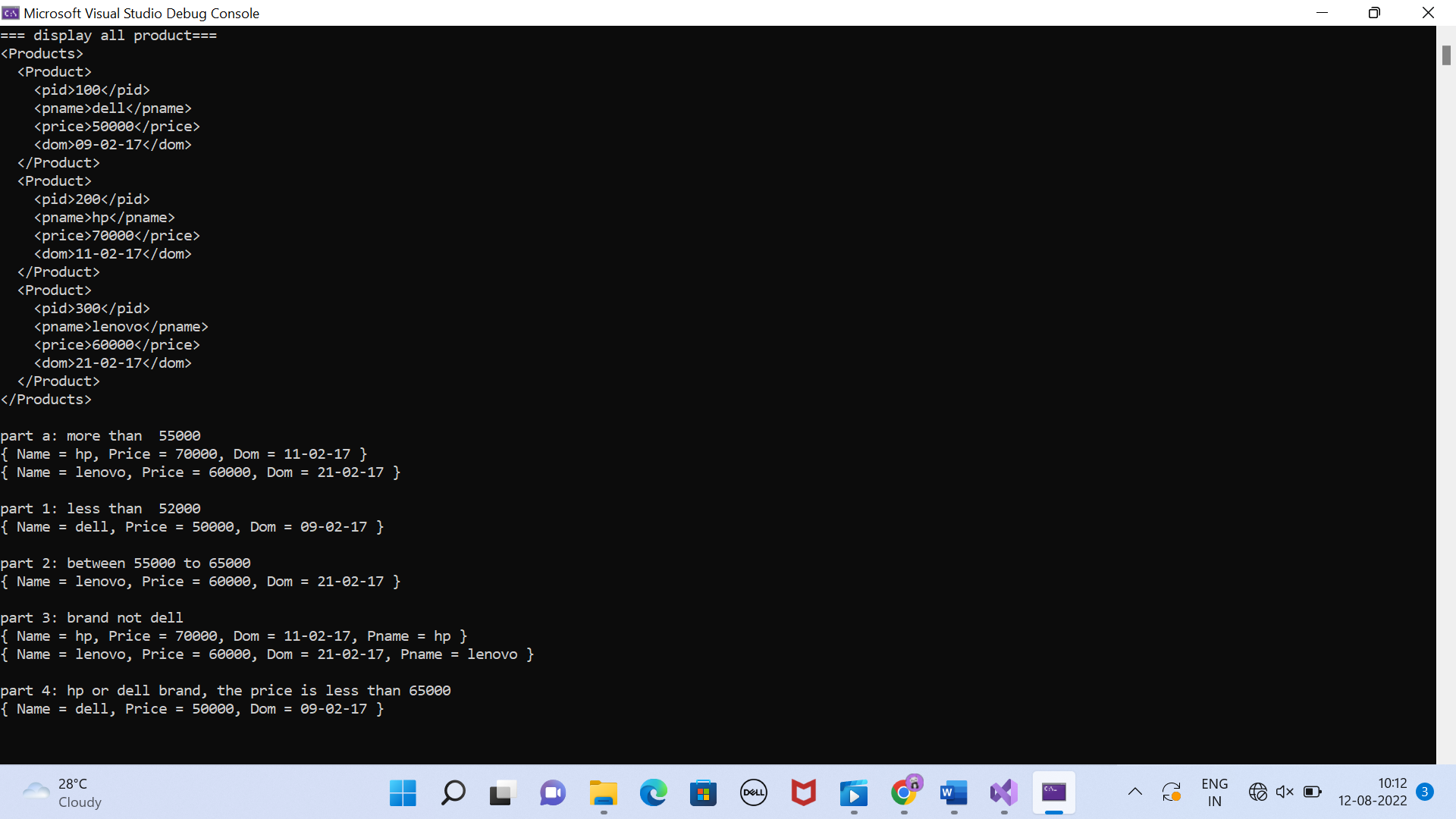
Console.WriteLine();

}

}

}

**OUTPUT:**



PART 4

**INPUT:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Xml.Linq;

using System.Threading.Tasks;

namespace pracs5sk

{

internal class Program

{

static void Main(string[] args)

{

XDocument pd = XDocument.Load("C:\\Users\\shruti\\source\\repos\\pracs5sk\\pracs5sk\\XMLFile1.xml");

var product = pd.Elements();

Console.WriteLine("part 4: hp or dell brand, the price is less than 65000");

var result4 = product.Elements("Product") //important line

.Where(x => (x.Element("pname").Value == "hp" || x.Element("pname").Value == "dell") && (int)x.Element("price") < 65000) // specifing data type is necessary

.Select(x => //generic

new //to access more than one column

{

Name = x.Element("pname").Value,

Price = x.Element("price").Value,

Dom = x.Element("dom").Value,

}

);

foreach (var r in result4)

{

Console.WriteLine(r.ToString());

}

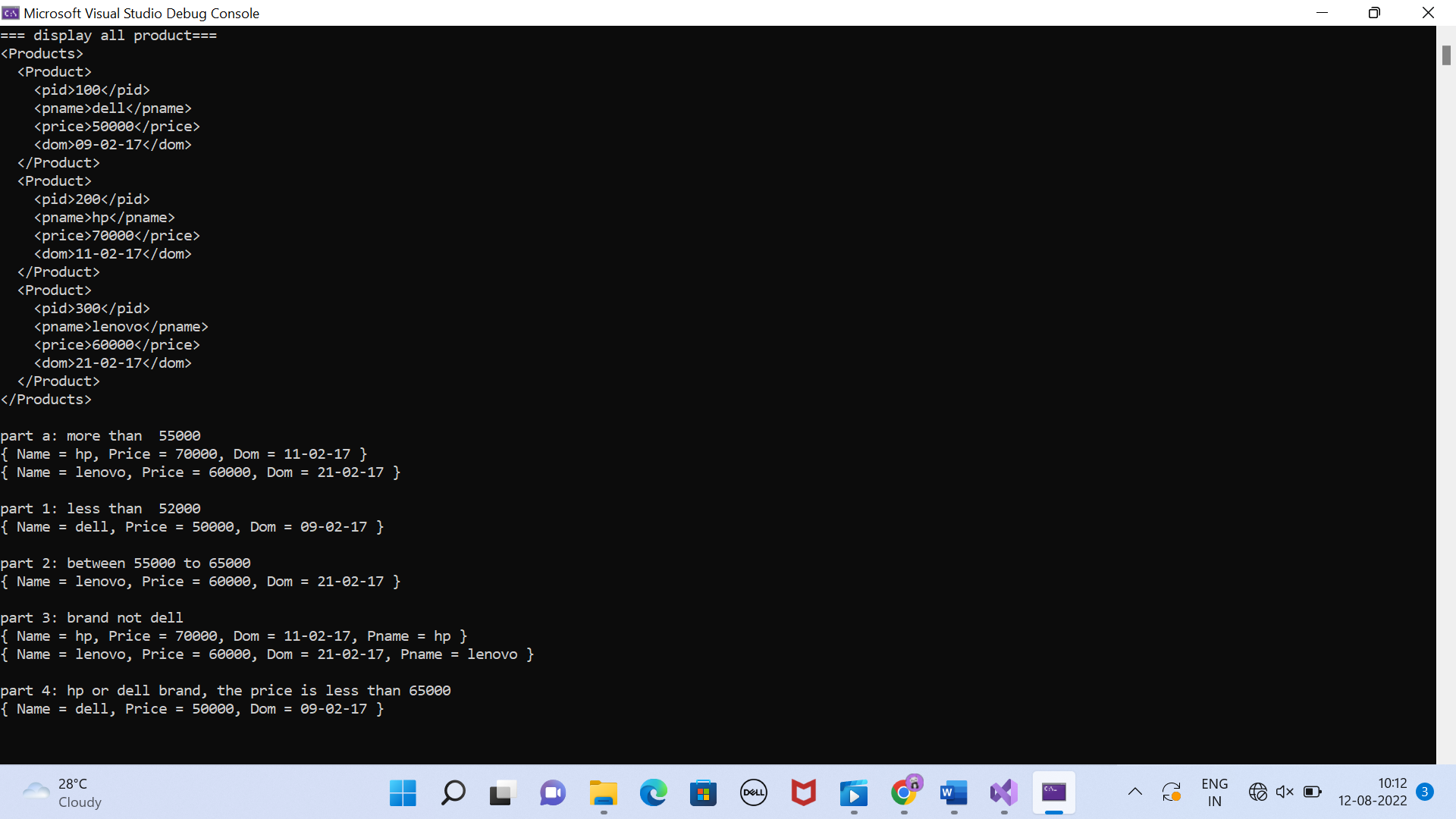
Console.WriteLine();

}

}

}

**OUTPUT:**



**PRACTICAL 6**

PART 1

(Display and salary of production department)

**INPUT:**

**OUTPUT:**

PART 2

(Combined salary of each department with their names)

**INPUT:**

**OUTPUT:**

PART 2

(Highest salary of each department along with their employee’s name)

**INPUT:**

**OUTPUT:**

**PRACTICAL 7**

(Razor page)

**INPUT:**using System.ComponentModel.DataAnnotations;

namespace pracs7skdb.Models

{

public class Employee

{

public int Id { get; set; }

//compulsory primary key, without this error will pop up

//this will not be displayed in ui so we put employee no

//case sensitive so make sure to keep i caps

//if you want to make any other as key so you add library 'using System.ComponentModel.DataAnnotations'

[StringLength(100,MinimumLength =3)] //more than one validation can be done

public String name { get; set; }

public int eno { get; set; }

public String dept { get; set; }

public int MyProperty { get; set; }

public int Salary { get; set; }

[EmailAddress] //these square brackets do automatic validation

public String email { get; set; }

[Phone]

public int phone { get; set; }

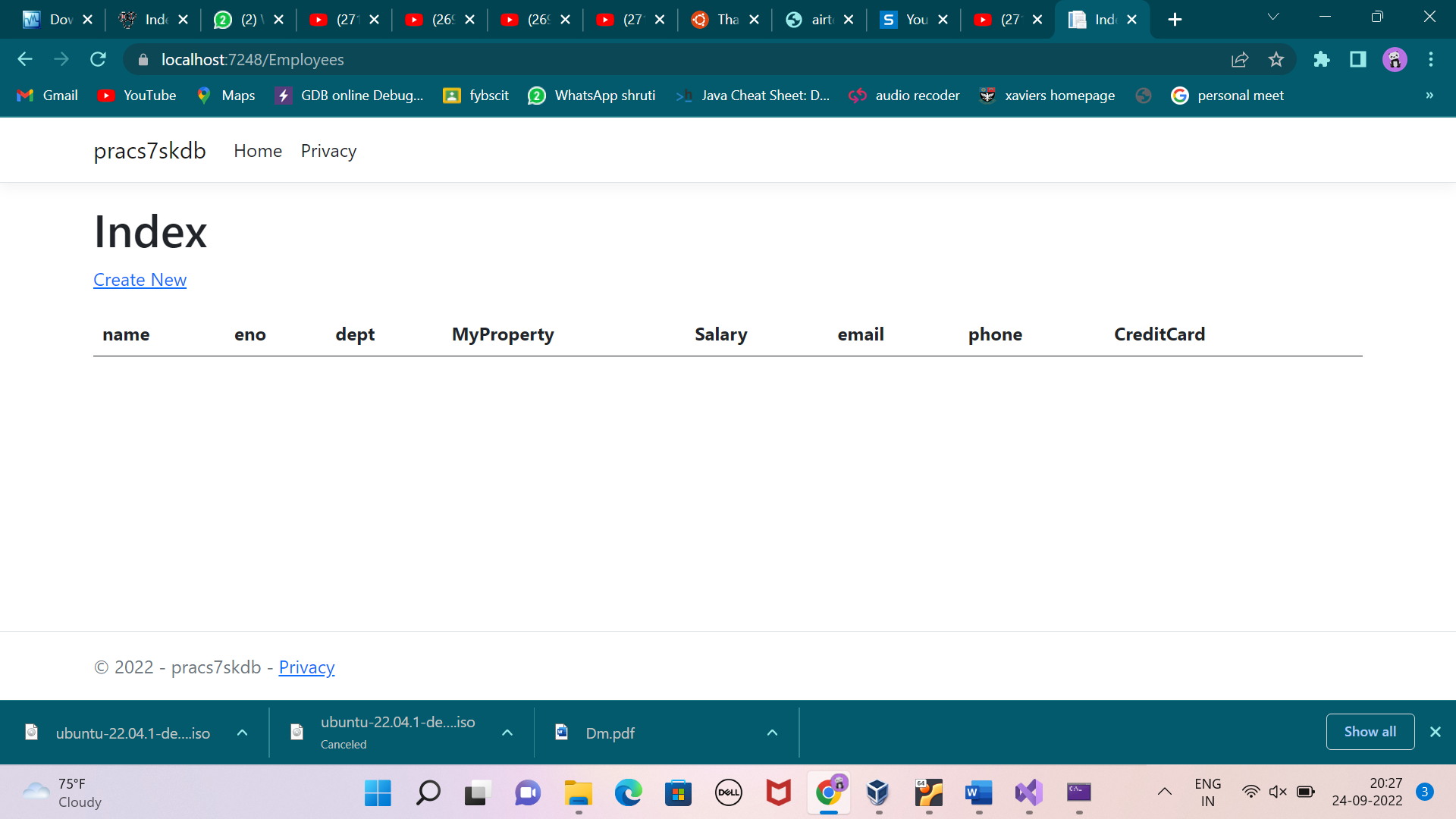
[CreditCard]

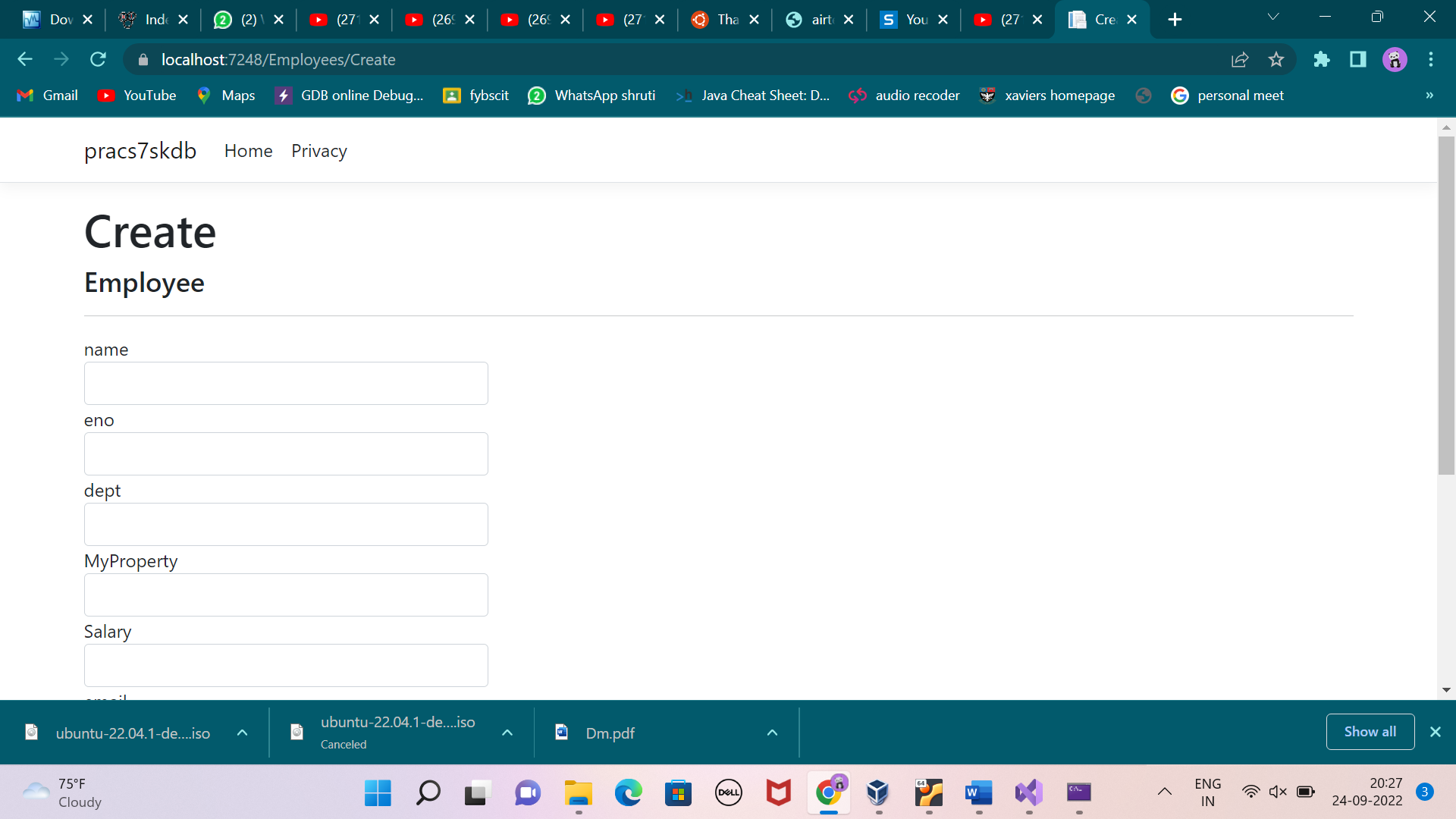
public int CreditCard { get; set; }

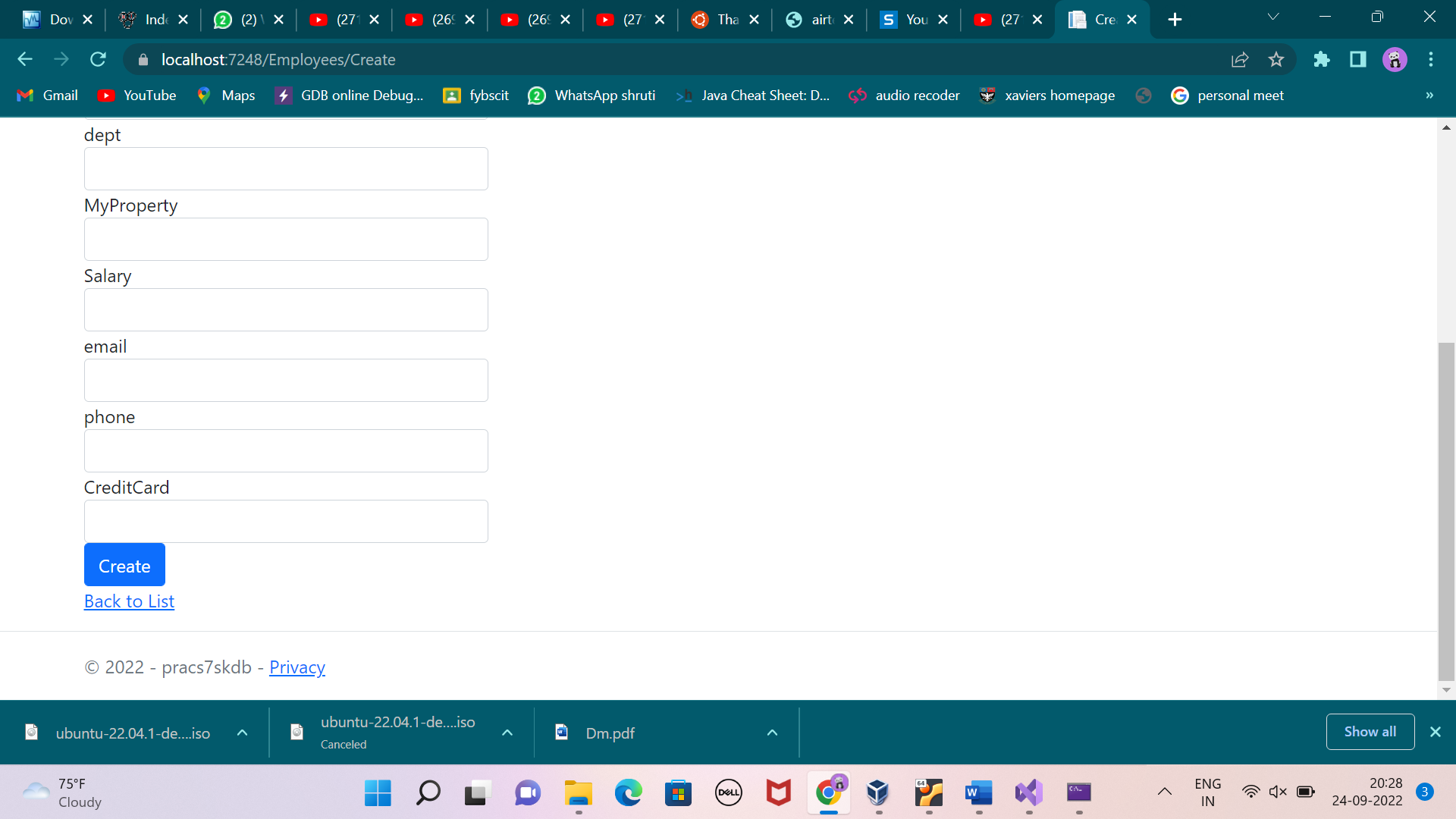
}

}

**OUTPUT:**







**PRACTICAL 8**

PART 1

(decimal to binary)

**INPUT:**

namespace sk\_pracs\_8

{

internal class Program

{

static void Main(string[] args)

{

/// decimal to binary

int n,i;

int[] arr = new int[10];

Console.WriteLine("enter decimal no.");

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

for( i=0; n>0; i++)

{

arr[i] = n % 2;

n = n / 2;

}

Console.WriteLine("binary form will be:");

for( i=i-1; i>=0; i--)

{

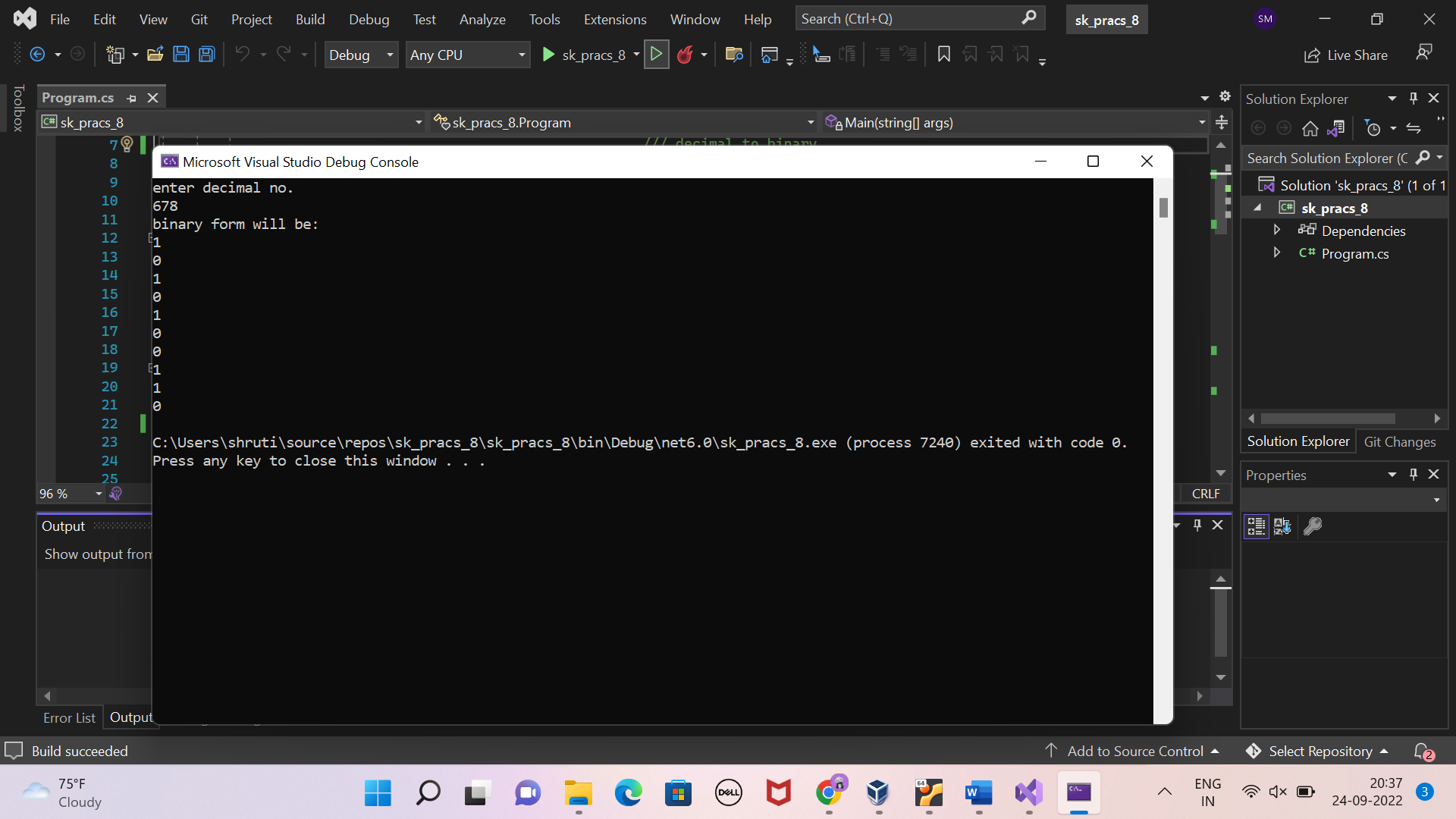
Console.WriteLine(arr[i]);

}

}

}

}



PART 2

(sum of digits of given number)

**INPUT:**

namespace sk\_pracs\_8

{

internal class Program

{

static void Main(string[] args)

{

/// sum of digits of given number

int n,sum=0,rem;

Console.WriteLine("number:");

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

while (n > 0)

{

rem = n % 10;

sum= sum + rem;

n = n / 10;

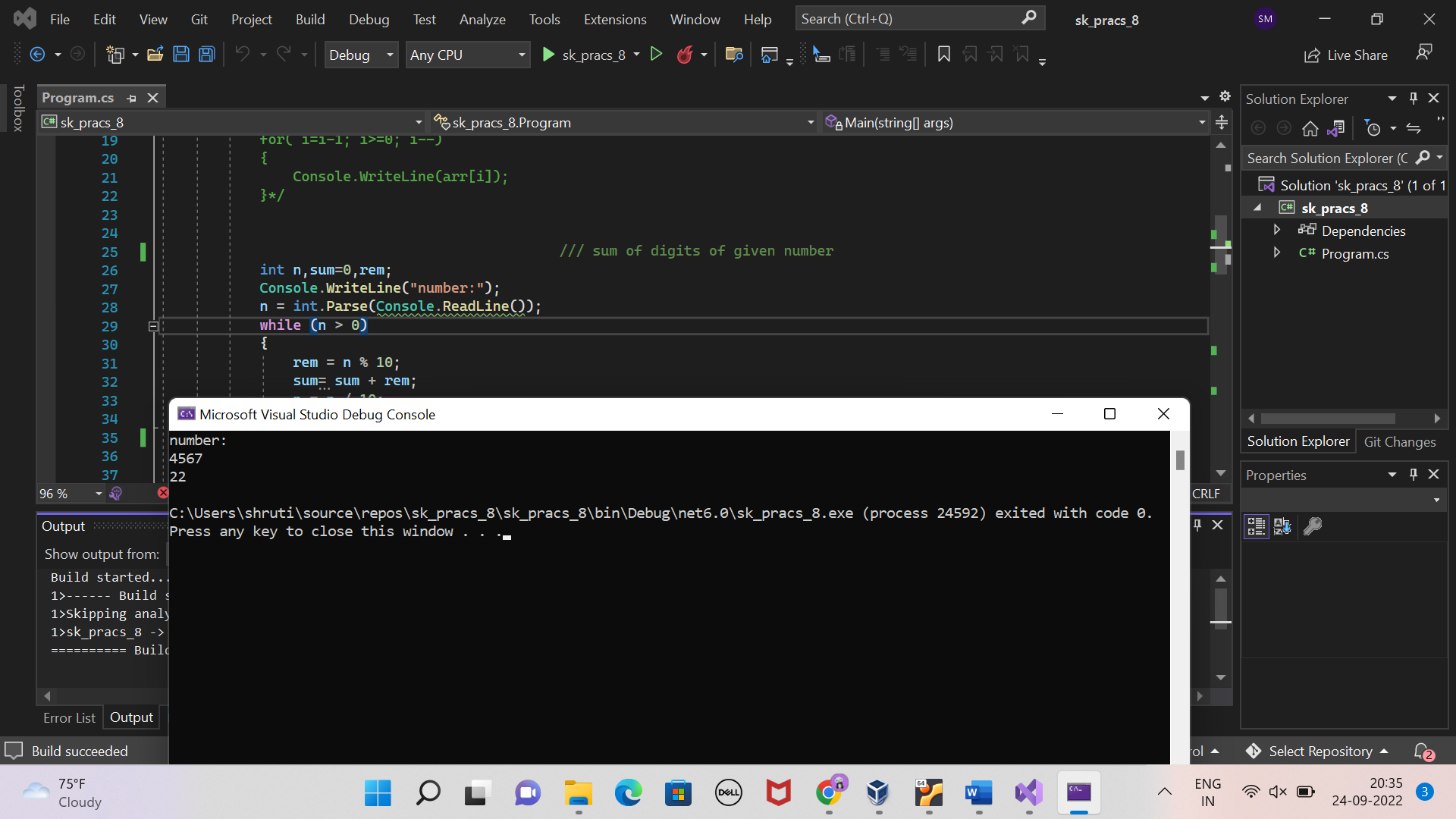
}

Console.WriteLine(sum);

}

}

}



PART 3

(palindrome)

**INPUT:**

namespace sk\_pracs\_8

{

internal class Program

{

static void Main(string[] args)

{

// / palindrome

int n, rev = 0, rem,temp;

Console.WriteLine("number pls:");

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

temp = n;

while (n > 0)

{

rem = n % 10;

rev = (rev \* 10) + rem;

n = n / 10;

}

if (rev == temp)

{

Console.WriteLine("palindrome");

}

else

{

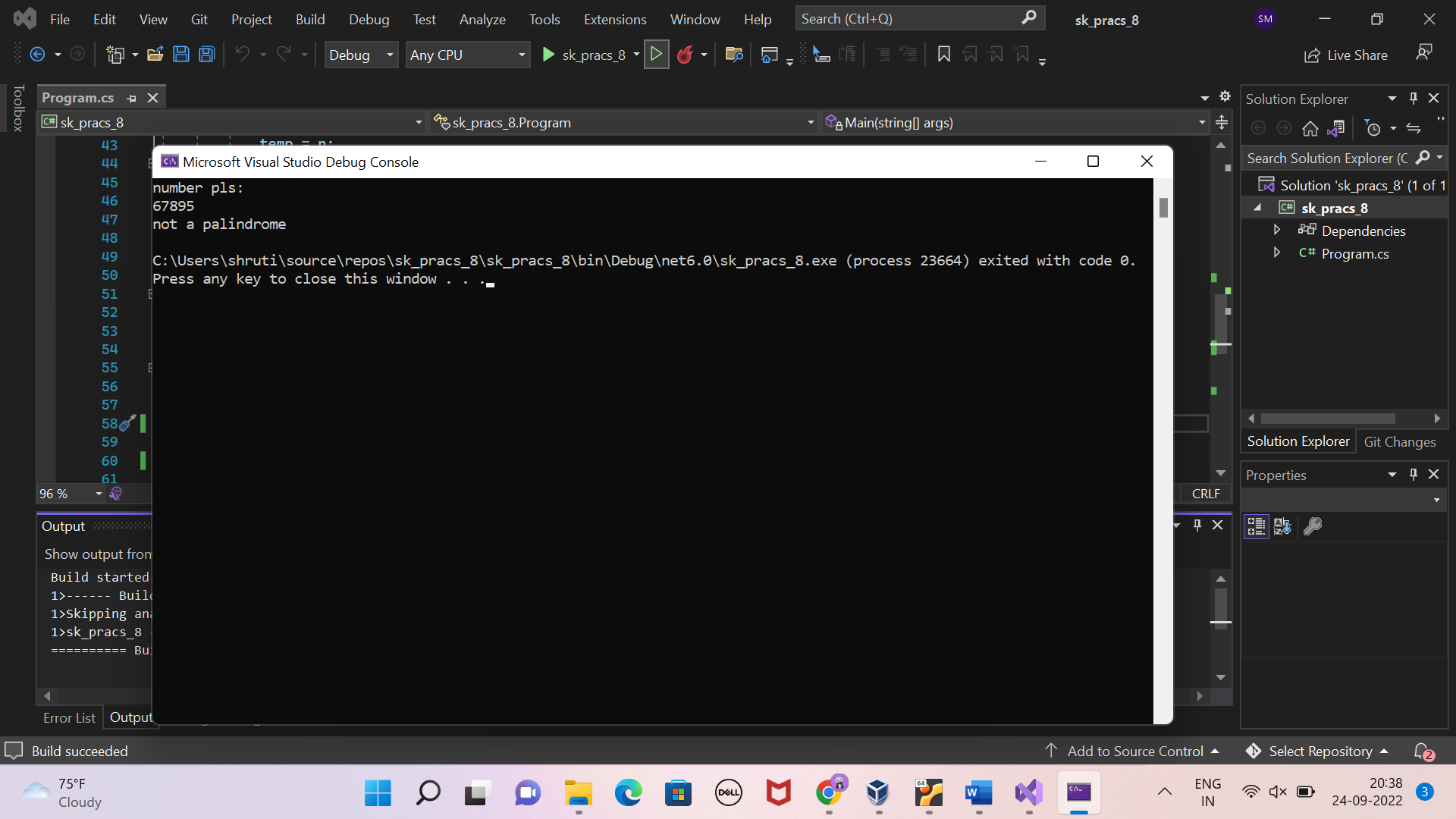
Console.WriteLine("not a palindrome");

}

}

}

}



namespace sk\_pracs\_8

{

internal class Program

{

static void Main(string[] args)

{

// fibonacci series

int n0=0, n1=1, n,k;

Console.WriteLine("limit");

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

k = k - 2;

Console.WriteLine(n0+"\n"+ n1);

while (k-- > 0)

{

n = n0 + n1;

n0 = n1;

n1 = n;

Console.WriteLine(n);

}

}

}

}

