Assignment-4

Oops

Student name: Pragati Gupta

Roll no: 0801CS201068

Question-1

import java.util.\*;

class Special

{ int N;

 int SumNumber;

 int original\_N;

    Special()

    {

        N=0;

        original\_N=0;

    }

    Special(int N)

    {

        this.N=N;

          original\_N=N;

    }

    public int  fact(int a)

    { int fact=1;

       while(a!=0)

       {

              fact\*=a;

              a-=1;

       }

       return fact;

    }

    public void sum()

    {

       while(N!=0)

       {

           int rem;

           rem=N%10;

           N=N/10;

          SumNumber+=this.fact(rem);

       }

    }

    public void isSpecial()

    { this.sum();

        if(this.SumNumber==original\_N)

        {

            System.out.println(original\_N+" is the special Number");

        }

        else

        {

            System.out.println(original\_N+"is not a special Number");

        }

    }

}

class Q1

{

    public static void main(String [] args)

    { Scanner s=new Scanner(System.in);

     int n=s.nextInt();

        Special A=new Special(n);

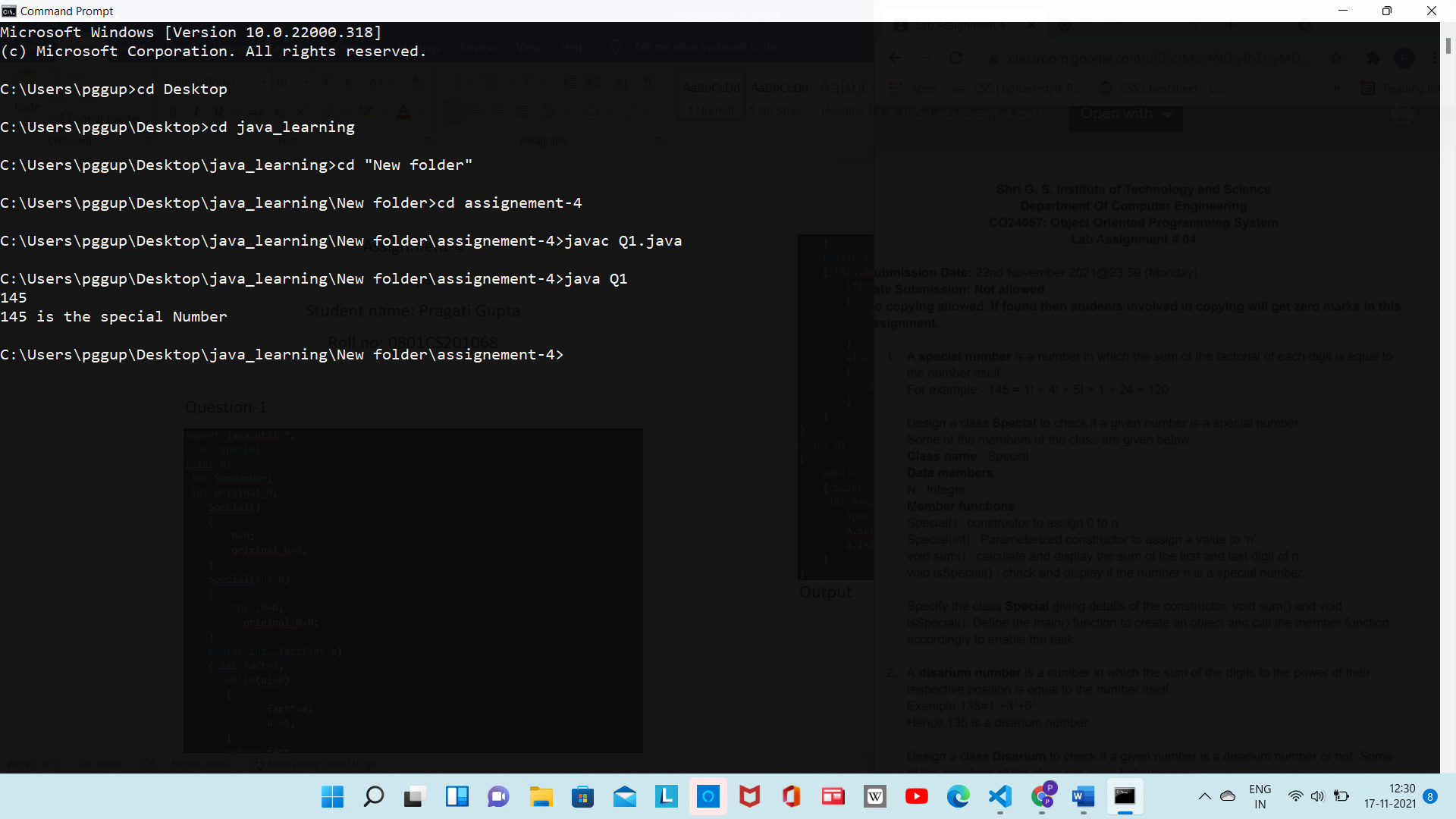
        A.sum();

        A.isSpecial();

    }

}

Output



Question-2

import java.util.\*;

class Disarium

{

    int num;

    int size;

   int sum;

   int original\_N;

    Disarium(int nn)

    {

        num=nn;

        size=0;

        original\_N=nn;

    }

    public int countDigit()

    {

        if(num==0)

        {

            return size;

        }

        else

        {  num/=10;

           size++;

            return countDigit();

        }

    }

    public void check()

    { int rem;

      num=original\_N;

      //System.out.println(num);

        while(num!=0)

        {  int result=1;

            rem=num%10;

            for(int i=size; i>0; i--)

            {

             result\*=rem;

            }

            sum+=result;

            size--;

             num/=10;

        }

        if(original\_N==sum)

        {

            System.out.println(original\_N+" is a Disarium number");

        }

        else

        {

            System.out.println(original\_N+" is not a Disarium number");

        }

    }

}

class Q2

{

    public static void main(String [] args)

    { Scanner s=new Scanner(System.in);

      int n=s.nextInt();

       Disarium a=new Disarium(n);

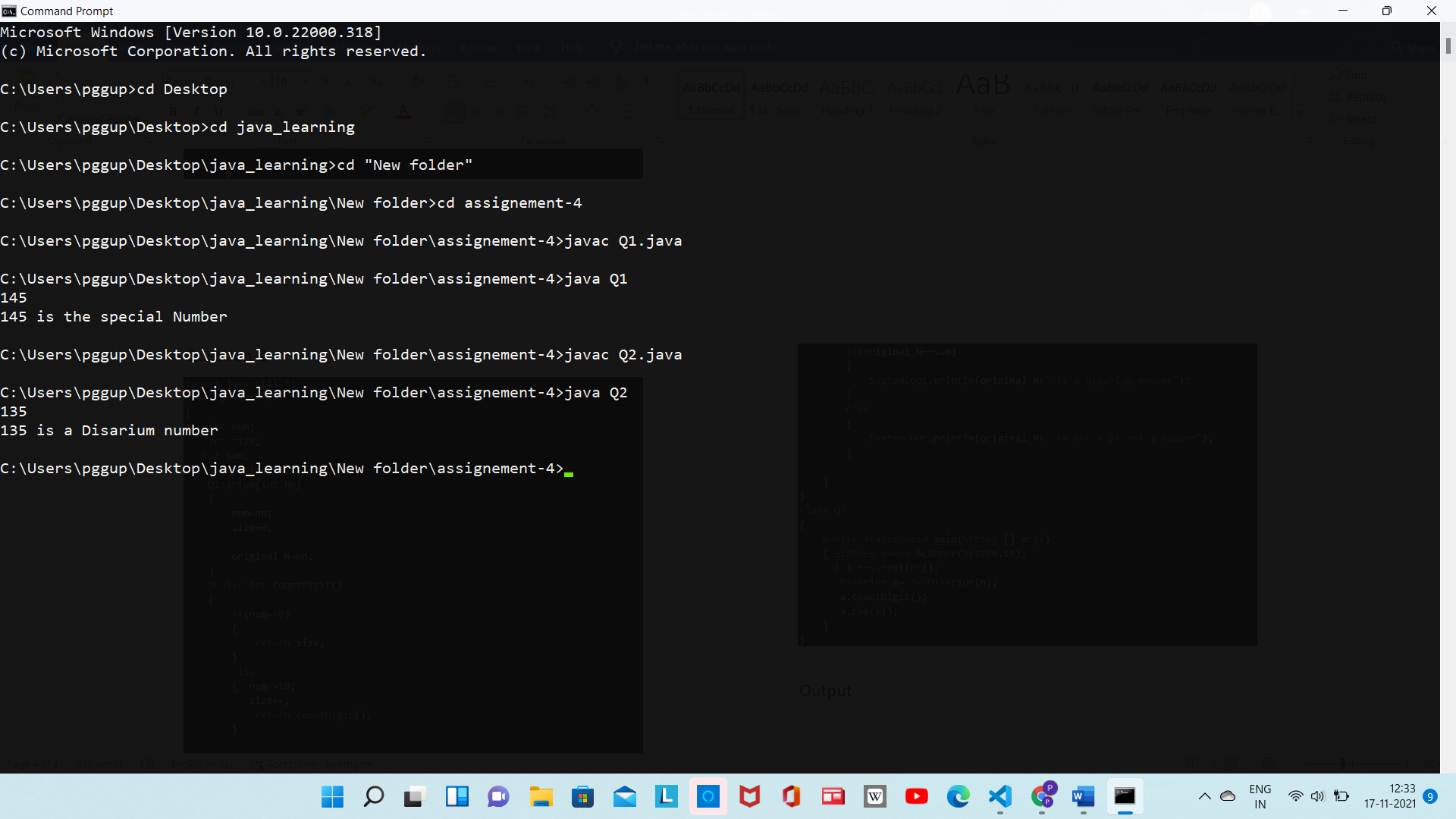
       a.countDigit();

       a.check();

    }

}

Output



Question-3

import java.util.\*;

class BinaryConverter

{

    int number;

    long binNumber;

  BinaryConverter(int number){

     this.number=number;

  }

    public long convert()

    {    int b=1;

        while(number!=0)

       {

           int rem=number%2;

           number=number-rem;

           number/=2;

           binNumber=binNumber+rem\*b;

              b\*=10;

       }

          return binNumber;

    }

    public void Display()

    { this.convert();

        System.out.println(" Given no. has a binary number :"+binNumber);

    }

}

class Q3

{

    public static void main(String [] args)

    {

            Scanner s=new Scanner(System.in);

            int n=s.nextInt();

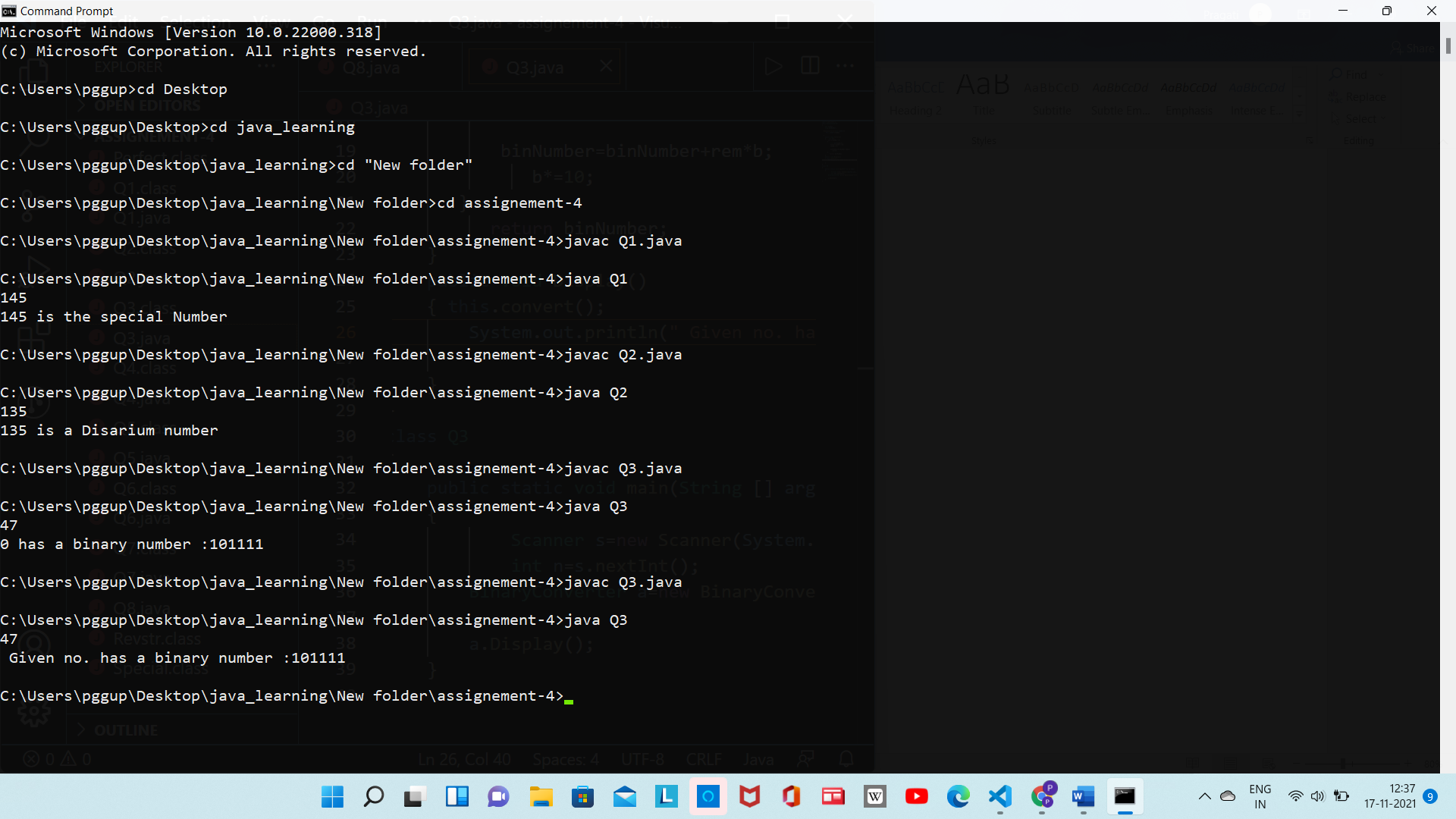
        BinaryConverter a=new BinaryConverter(n);

        a.Display();

    }

}

Output



Question-4

import java.util.\*;

class Q4

{

  public static void main(String [] args)throws Exception

  {

       Scanner s=new Scanner(System.in);

       int n=s.nextInt();

      Emrip a=new Emrip(n);

      a.isEmrip();

  }

}

class Emrip

{

    int n;

    int rev;

    int f;

    Emrip(int nn)

    {

        n=nn;

        rev=0; f=2;

    }

    public int isPrime(int x)

    { int i=2;

        while(x%i!=0 )

        {

            i++;

        }

        if(i==x)

        {

            return 1;

        }

        else

        {

            return 0;

        }

    }

    public void isEmrip()

    {  int result1=this.isPrime(n);

        int b=1;

        if(result1==1)

        {

        while(n!=0 )

        {

        int rem=n%10;

        n/=10;

        rev=rev\*b+rem;

        b\*=10;

        }

        }

        int result=this.isPrime(rev);

        if(result1==1)

        { System.out.println("Given No. is prime");

         System.out.println( rev+" reverse of a given number is a prime number");

        }

         else if (result==0)

       {

            System.out.println("Given No. is prime");

         System.out.println( rev+" reverse of a given number is not a prime number");

        }

       else{

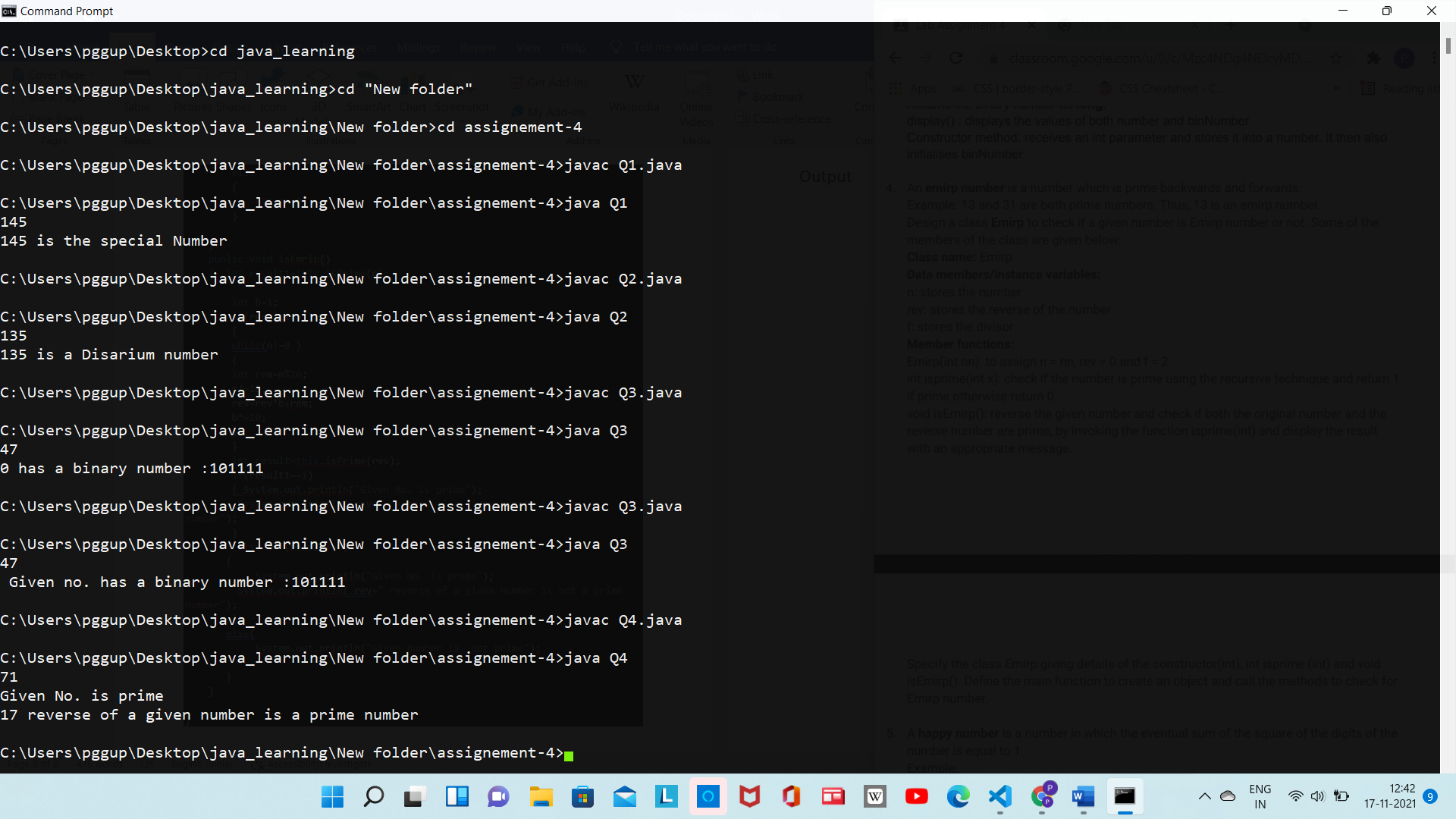
            System.out.println("Given number is  not prime");

       }

    }

}

Output



Question-5

import  java.util.\*;

class Happy

{

    int n;

    int sum;

    Happy()

    {

         n=0;

         sum=0;

    }

    public void getnum(int nn)

    {

          n=nn;

    }

    public int sum\_sq\_digits(int x)

    {

        if (x==0)

        {

            return sum;

        }

        else

        {

           int rem=x%10;

           x/=10;

           sum +=(rem\*rem);

            return sum\_sq\_digits(x);

        }

    }

    public void ishappy()

    {

       this.sum\_sq\_digits(n);

       if(sum==n)

       {

           System.out.println(n+" is find be a Happy number");

       }

       else

       {

           System.out.println(n+" is not a  Happy number ");

       }

    }

}

class Q5

{

    public static void main(String [] args)

    {

        Scanner a=new Scanner(System.in);

        int Qn=a.nextInt();

        Happy A=new Happy();

        // Happy [] B=new Happy[100] ;

        A.getnum(Qn);

        A.ishappy();

        // for (int i=1; i<100; i++)

        // {

        //      B[i]=new Happy();

        //     B[i].getnum(i);

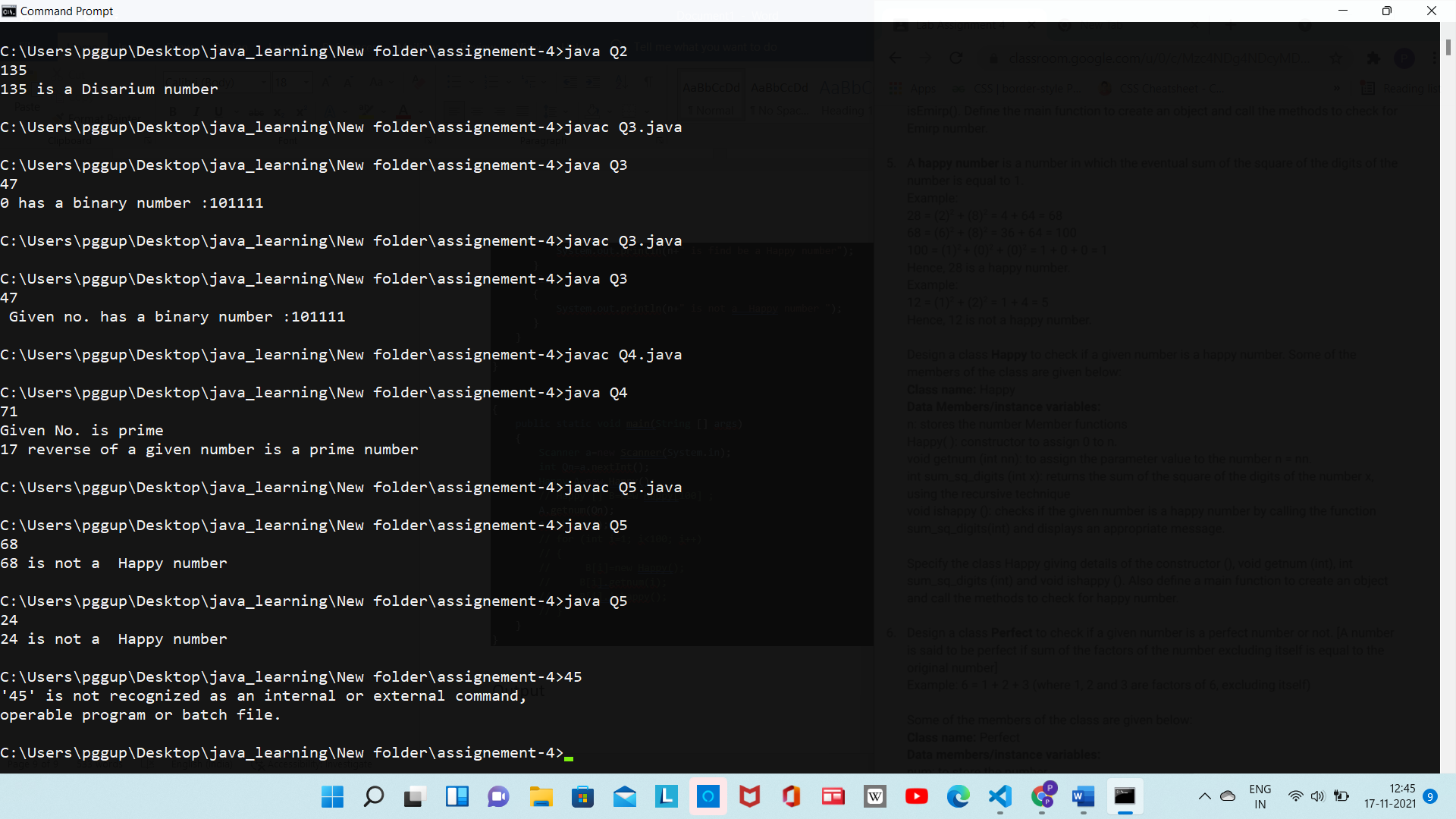
        //     B[i].ishappy();

        // }

    }

}

Output



Question-6

import java.util.\*;

class Perfect

{

    int num;

    int sum;

    Perfect(int nn)

    {

        num=nn;

        sum=0;

    }

    public int sum\_of\_factors(int i)

    { int a=1;

        if ((i/2-a)==0)

         return sum;

         else

           { int fact=1;

               sum+= (i%a==0 ? i : 0);

               a++;

               fact\*=a;

               return sum\_of\_factors(i/fact);

           }

        }

        public void check()

        { this.sum\_of\_factors(num);

           if(sum==num)

           {

               System.out.println(num +" is a perfect number");

           }

           else

             System.out.println(num +" is not a perfect number");

        }

}

class Q6

{

    public static void main(String [] args)

    {

        Scanner s=new Scanner(System.in);

        int n=s.nextInt();

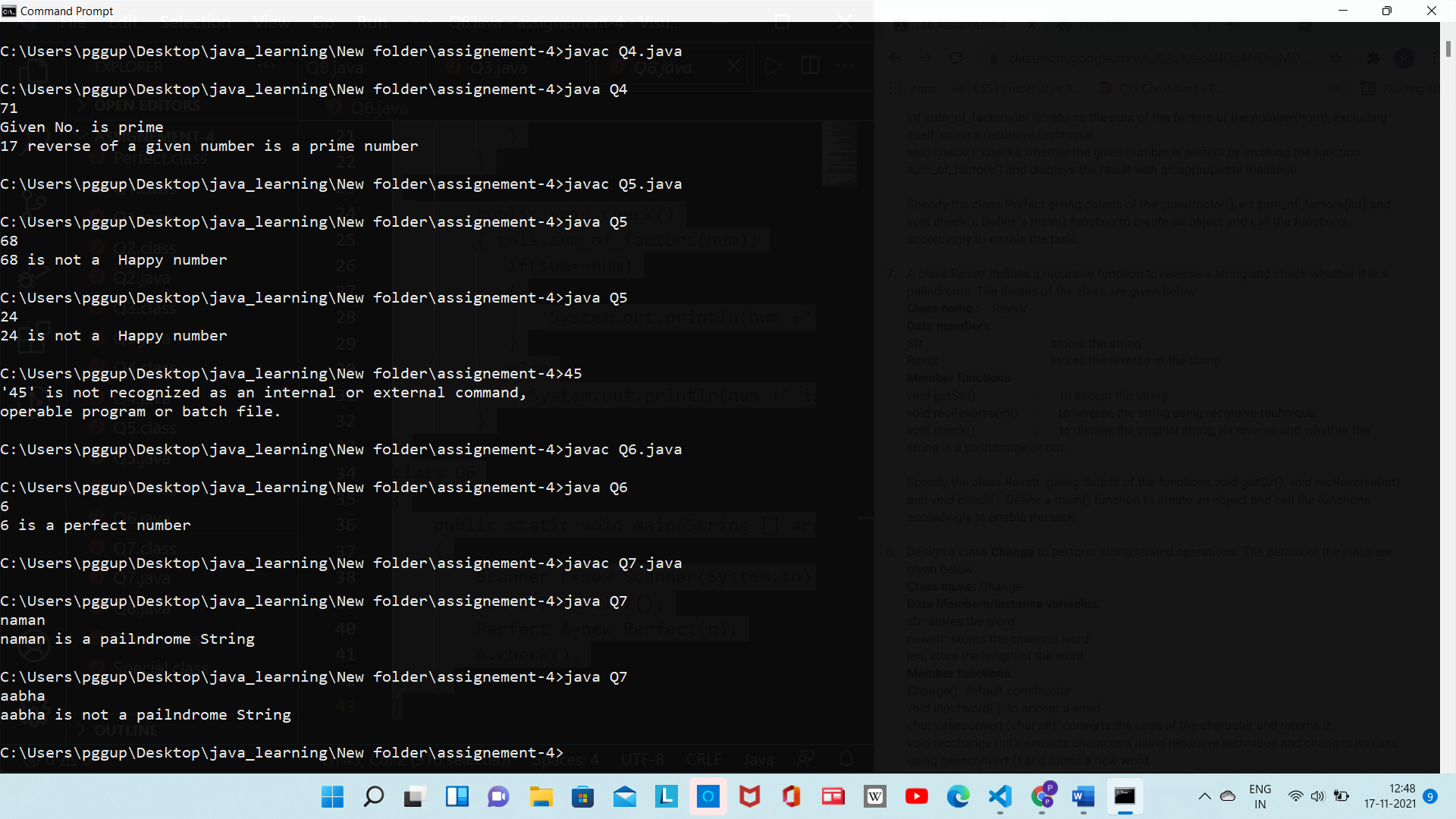
        Perfect A=new Perfect(n);

        A.check();

    }

}

Output



Question-7

import java.util.\*;

class Revstr

{

    String Str;

    //String Revst;

    String Revst =new String();

   int length;

    public void getStr(String Str)

    {

       this.Str=Str;

       length=Str.length();

       //Revst=null;

    }

    String recReverse(int x)

    { int i=0;

        if(x==0)

        {

          return Revst;

        }

        else

        {  --x;

            Revst+=Str.charAt(x);

            i++;

            return recReverse(x);

        }

    }

    void check()

    {

        this.recReverse(length);

        if(Revst.equals(Str))

        {

            System.out.println(Str+" is a pailndrome String");

        }

        else

        {

             System.out.println(Str +" is not a pailndrome String");

        }

    }

}

class Q7

{

    public static void main(String [] args)

    {

        Scanner s=new Scanner(System.in);

        String b=s.nextLine();

        Revstr A=new Revstr();

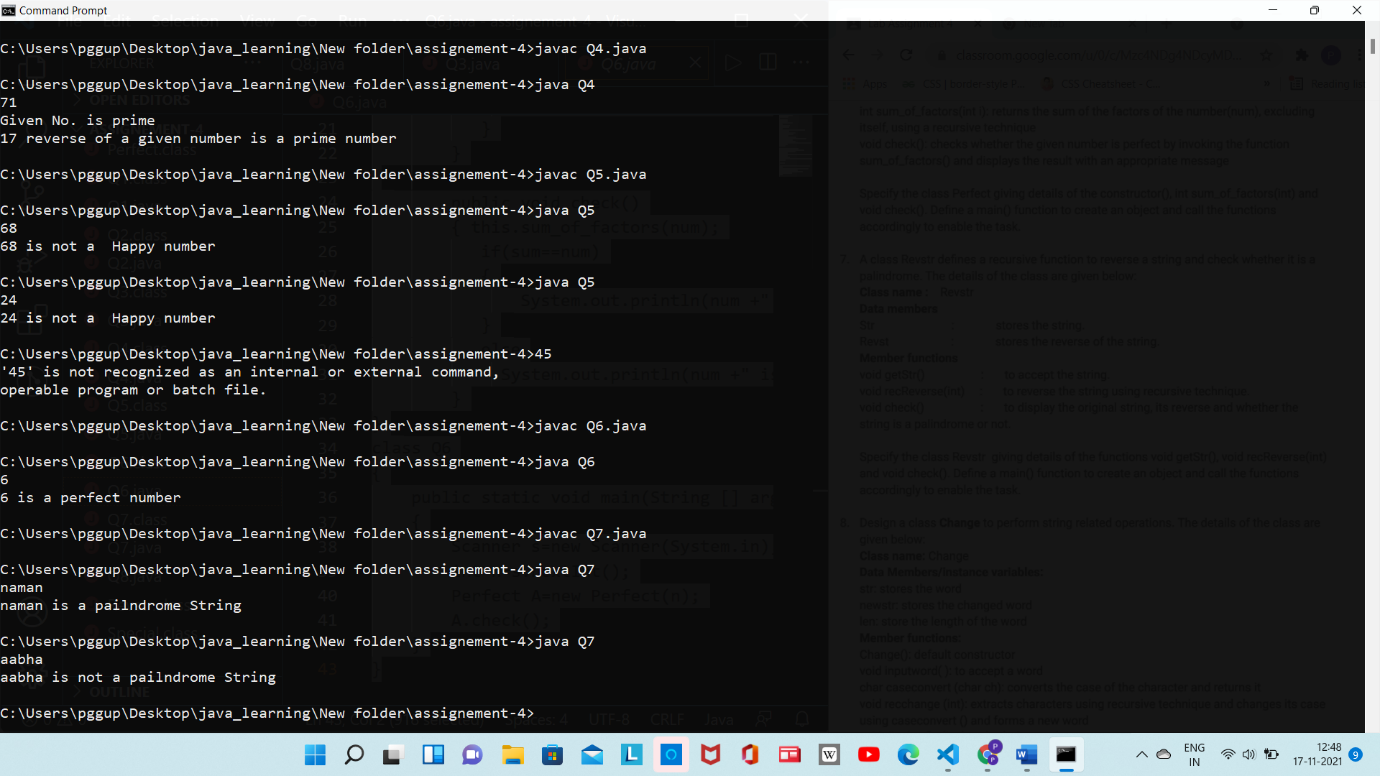
        A.getStr(b);

        A.check();

    }

}

Output



Question-7

import java.util.\*;

class Revstr

{

    String Str;

    //String Revst;

    String Revst =new String();

   int length;

    public void getStr(String Str)

    {

       this.Str=Str;

       length=Str.length();

       //Revst=null;

    }

    String recReverse(int x)

    { int i=0;

        if(x==0)

        {

          return Revst;

        }

        else

        {  --x;

            Revst+=Str.charAt(x);

            i++;

            return recReverse(x);

        }

    }

    void check()

    {

        this.recReverse(length);

        if(Revst.equals(Str))

        {

            System.out.println(Str+" is a pailndrome String");

        }

        else

        {

             System.out.println(Str +" is not a pailndrome String");

        }

    }

}

class Q7

{

    public static void main(String [] args)

    {

        Scanner s=new Scanner(System.in);

        String b=s.nextLine();

        Revstr A=new Revstr();

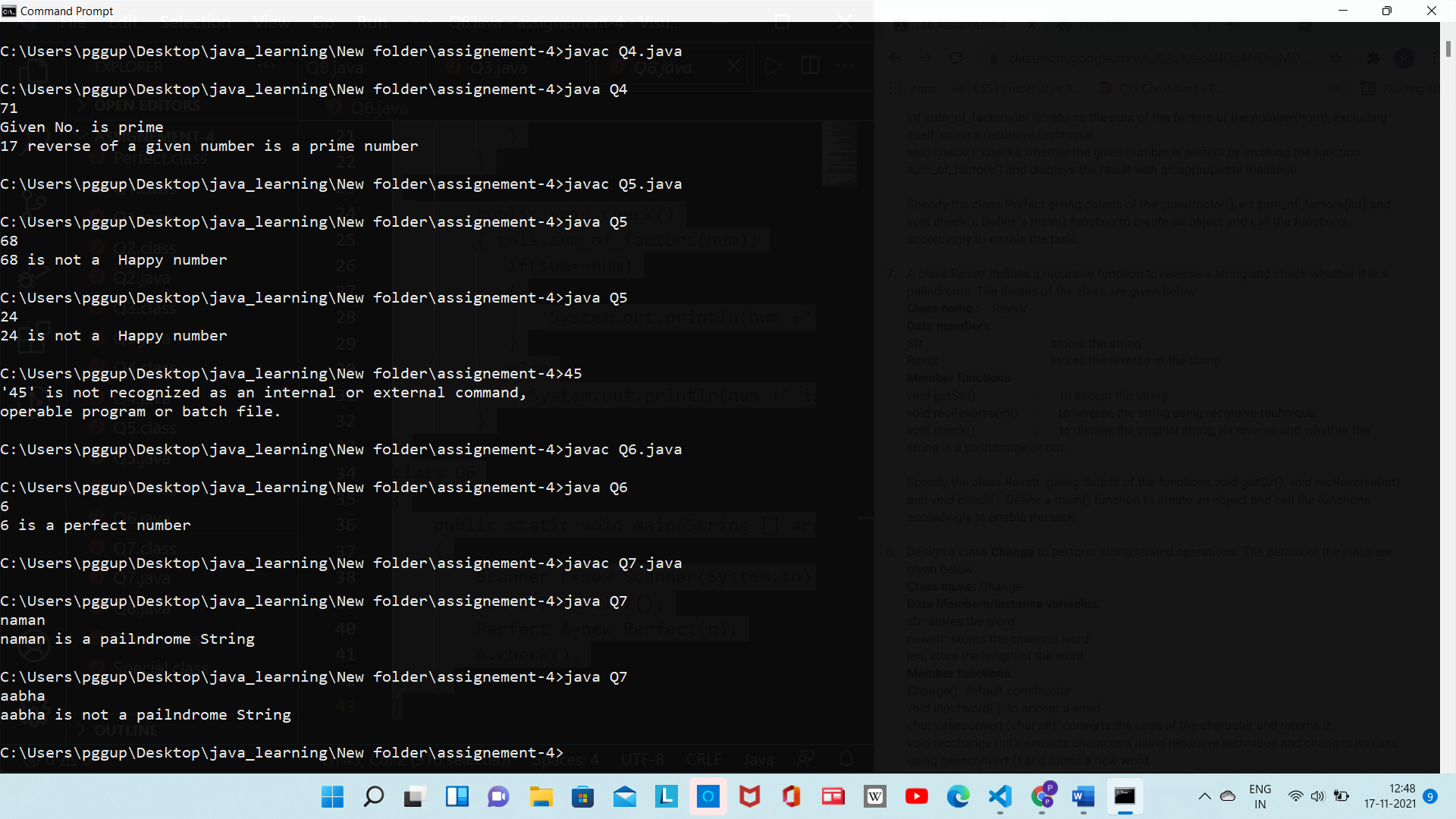
        A.getStr(b);

        A.check();

    }

}

Output



Question-8

import java.util.\*;

class Change

{  int j=0;

    String str;

    String newstr =new String();

    int len;

    Change()

    {

        str=null;

        len=0;

    }

 public void inputword(String x)

 {

      str=x;

      len=str.length();

 }

 public String recchange (int i)

 {

     if(i==0)

     {

         return newstr;

     }

     else

    {

       --i;

       char ch;

        ch=str.charAt(j);

        newstr+=this.caseconvert(ch);

        j++;

        return recchange(i);

    }

 }

 char caseconvert(char ch)

 {

     if(Character.isUpperCase(ch))

     {

         ch=Character.toLowerCase(ch);

         return ch;

     }

     else

     {

         ch=Character.toUpperCase(ch);

         return ch;

     }

 }

 void Display()

 {

      System.out.println("Length of the given String is :"+len);

     System.out.println("Old String is :"+str);

     System.out.println("New String is :"+newstr);

 }

}

class Q8

{

    public static void main(String [] args)

    {

           Scanner s=new Scanner(System.in);

           String str=s.nextLine();

           int n;

           Change a=new Change();

           a.inputword(str);

           System.out.println("How many word you like to convert");

            n=s.nextInt();

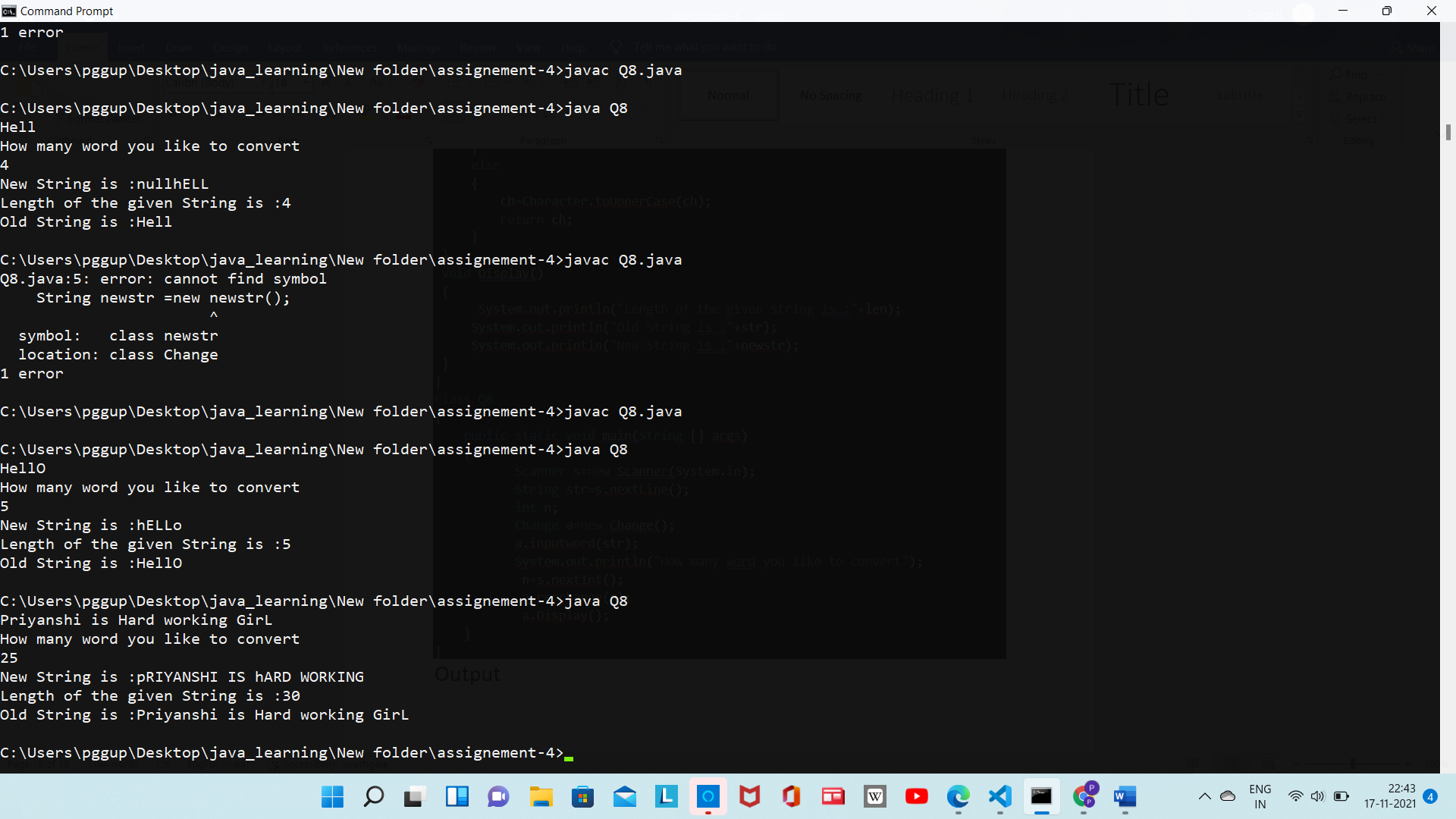
            a.recchange(n);

            a.Display();

    }

}

Output



Question-9

import java.util.\*;

 interface Shape

{ public static double pie=22/7;

    abstract double area();

}

class Circle implements Shape

{

  double radius;

    Circle (double radius )

    {

        this.radius=radius;

    }

    public double area()

    {

        return  pie\*radius\*radius;

    }

}

class Rectangle implements Shape{

    double Length;

    double breadth;

    Rectangle(double Length,double breadth)

    {

        this.Length=Length;

        this.breadth=breadth;

    }

    public double  area()

    {

        return Length\*breadth;

    }

}

class Q9

{

    public static void main(String [] args) throws Exception

    { Scanner in =new Scanner(System.in);

       double r,l,b;

        r=in.nextDouble();

        System.out.flush();

        l=in.nextDouble();

        b=in.nextDouble();

         Shape [] geoObject =new Shape[2];

         Shape A=new Circle(r);

         Shape B=new  Rectangle (l,b);

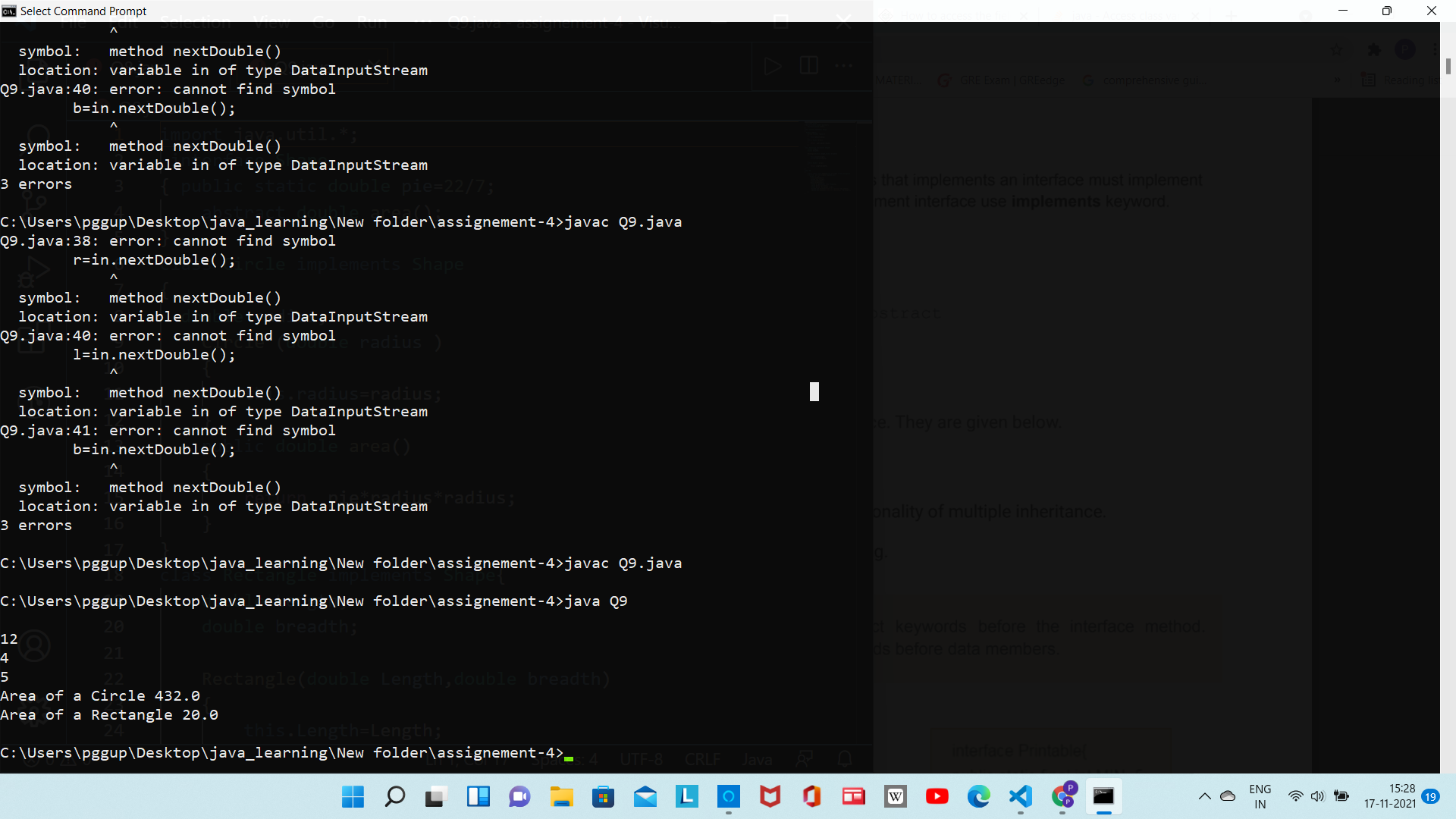
         System.out.println("Area of a Circle "+A.area());

         System.out.println("Area of a Rectangle "+B.area());

    }

}

Output



Question-10

import java.util.\*;

interface Data

{ public static final double pie=22/7;

    abstract double volume();

}

class Base

{

     double rad;

        Base(double rad)

        {

              this.rad=rad;

        }

        public void show()

        {

            System.out.println("The radius of the shape is :"+rad);

        }

}

class CalVol extends  Base implements  Data

{

   double ht;

    CalVol(double rad,double ht)

    { super(rad);

        this.ht=ht;

    }

    public double volume()

    {

        return pie\*rad\*ht;

    }

    public void show()

    {

        super.show();

        System.out.println("Height of the shape is :"+ht);

        System.out.println("Area of the Shape :"+this.volume());

    }

}

class Q10

{

    public static void main(String [] args) throws Exception

    {

        Scanner s=new Scanner(System.in);

        double r,h;

        r=s.nextDouble();

        h=s.nextDouble();

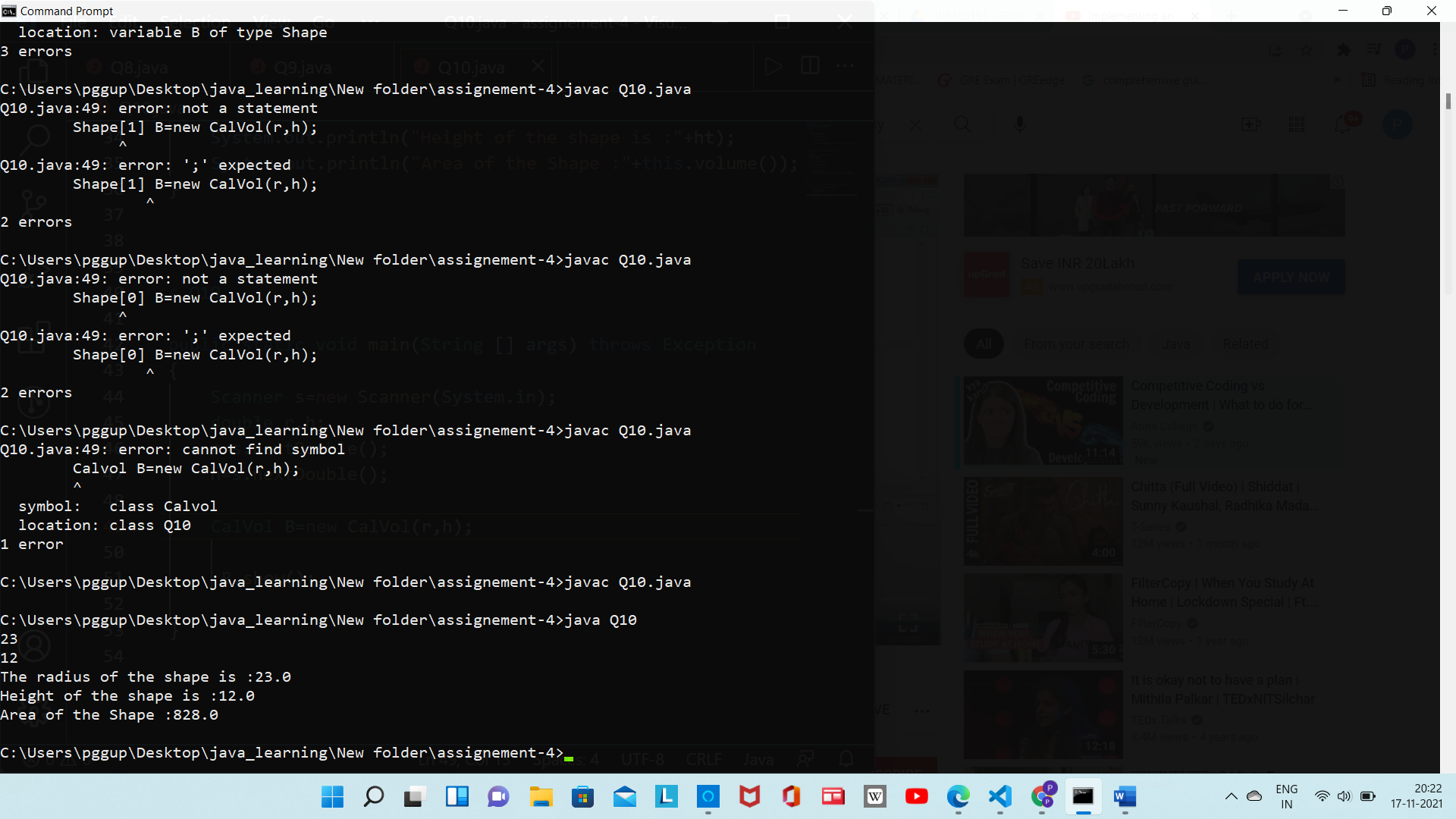
        CalVol B=new CalVol(r,h);

         B.show();

    }

}

Output



End…