**C-DAC Mumbai Date 25/09/2024**

**Subject: Algorithm and Data Structure**

**Assignment 1**

**Solve the assignment with following thing to be added in each question.**

-Program

-Flow chart

-Explanation

-Output

-Time and Space complexity

1. Armstrong Number

Problem: Write a Java program to check if a given number is an Armstrong number.

Test Cases:

Input: 153

Output: true

Input: 123

Output: false

import java.util.Scanner;

public class Armstrong{

public static boolean Armstrongval(int number) {

int num= number;

int digits = 0;

int sum = 0;

while (num != 0) {

num /= 10;

digits++;

}

num = number;

while (num != 0) {

int remainder = num % 10;

sum += Math.pow(remainder, digits);

num /= 10;

}

return sum == number;

}

public static void main(String[] args) {

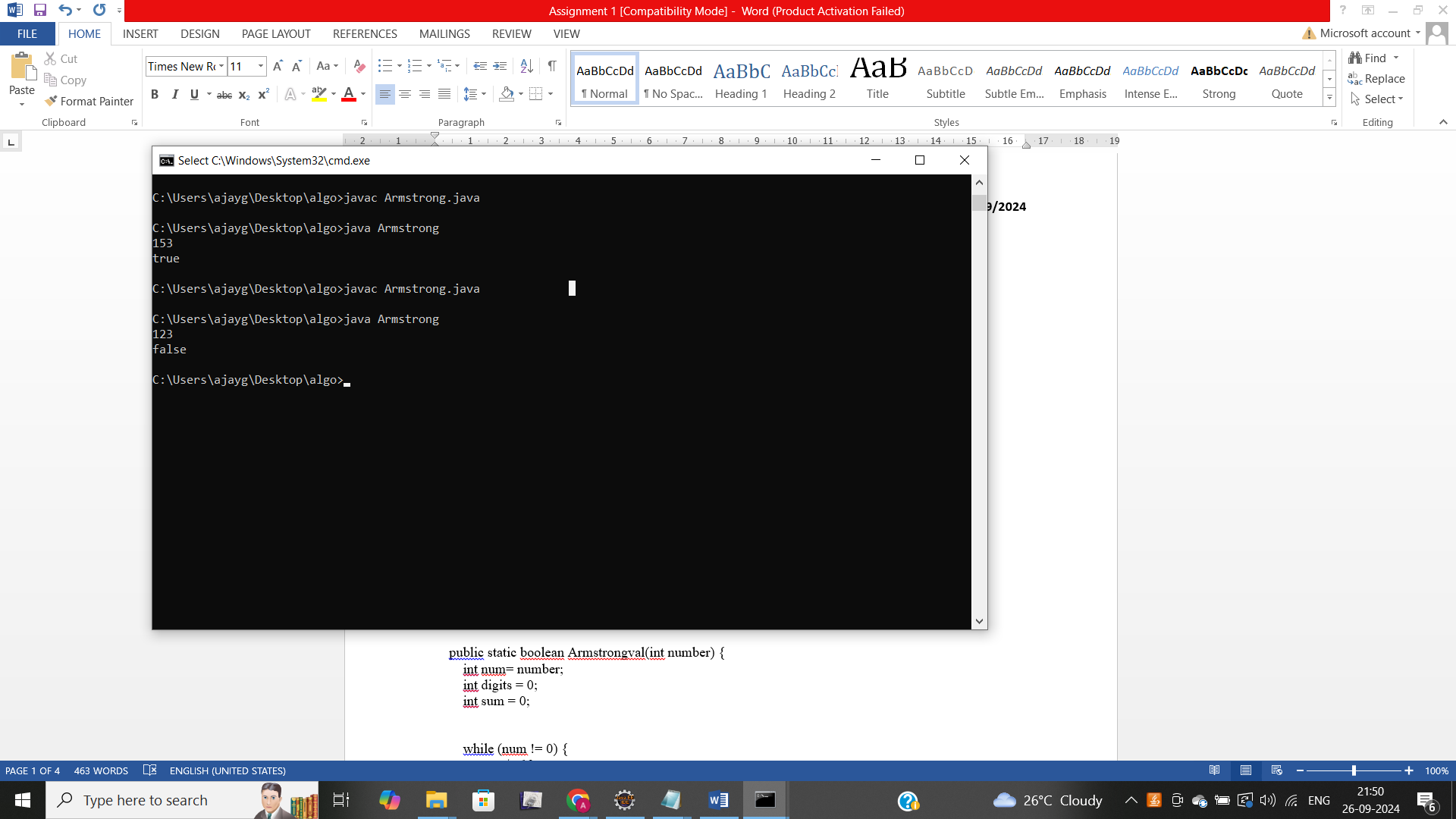
Scanner scanner = new Scanner(System.in);

int number = scanner.nextInt();

System.out.println(Armstrongval(number));

}

}



2. Prime Number

Problem: Write a Java program to check if a given number is prime.

Test Cases:

Input: 29

Output: true

Input: 15

Output: false

import java.util.Scanner;

class prime{

public static boolean isprime(int number, int divisor){

if(number<=1)

return false;

if(divisor\*divisor>number){

return true;

}

if(number % divisor==0){

return false;

}

return isprime(number,divisor+1);

}

public static void main(String [] args){

Scanner sc=new Scanner(System.in);

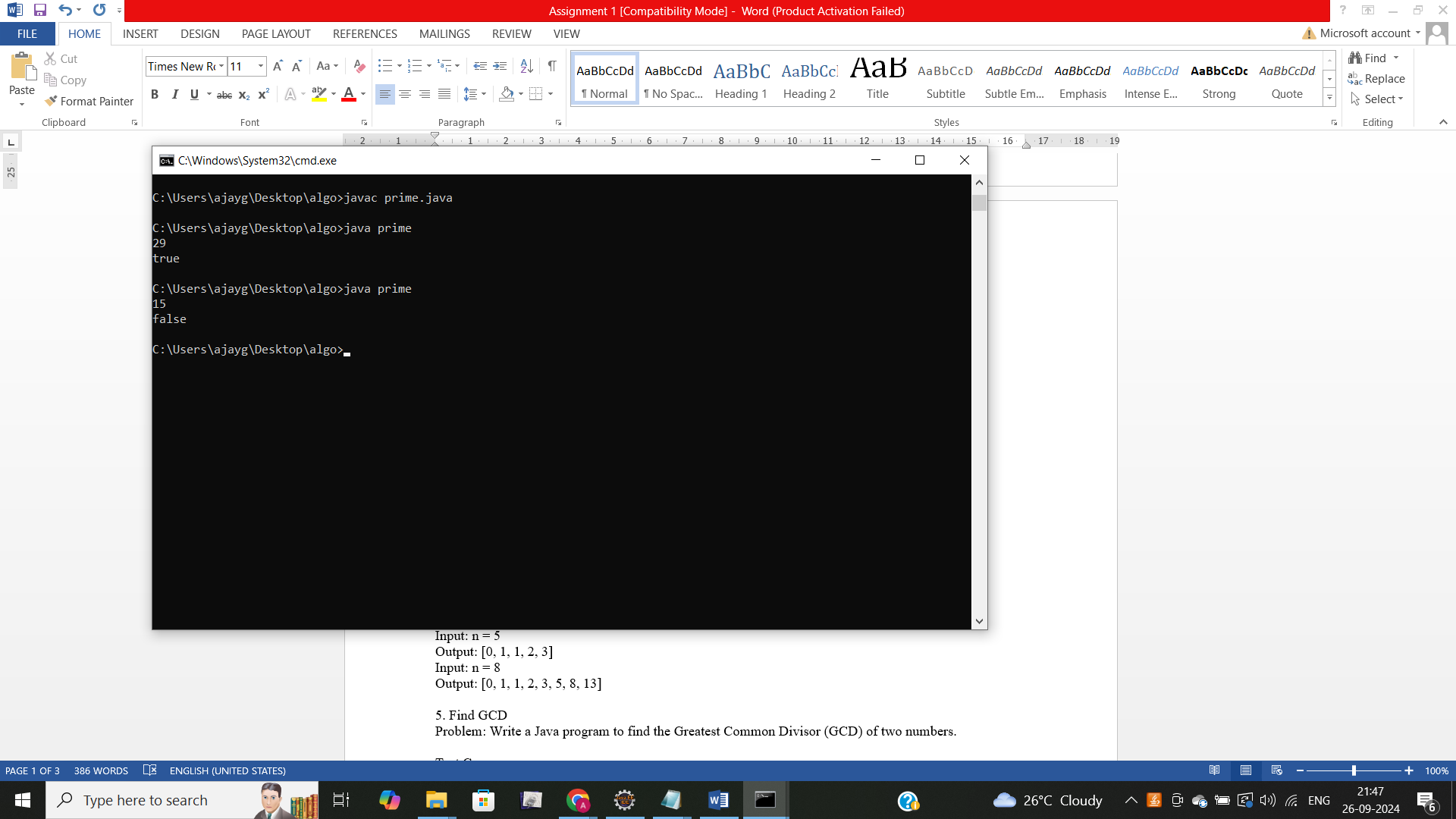
int number= sc.nextInt();

boolean result=isprime(number,2);

System.out.println(result);

}

}



3. Factorial

Problem: Write a Java program to compute the factorial of a given number.

Test Cases:

Input: 5

Output: 120

Input: 0

Output: 1

import java.util.Scanner;

class fact{

static int factorial(int num){

if(num<=1){

return 1;

}

else{

return num\*factorial(num-1);

}

}

public static void main(String [] args)

{

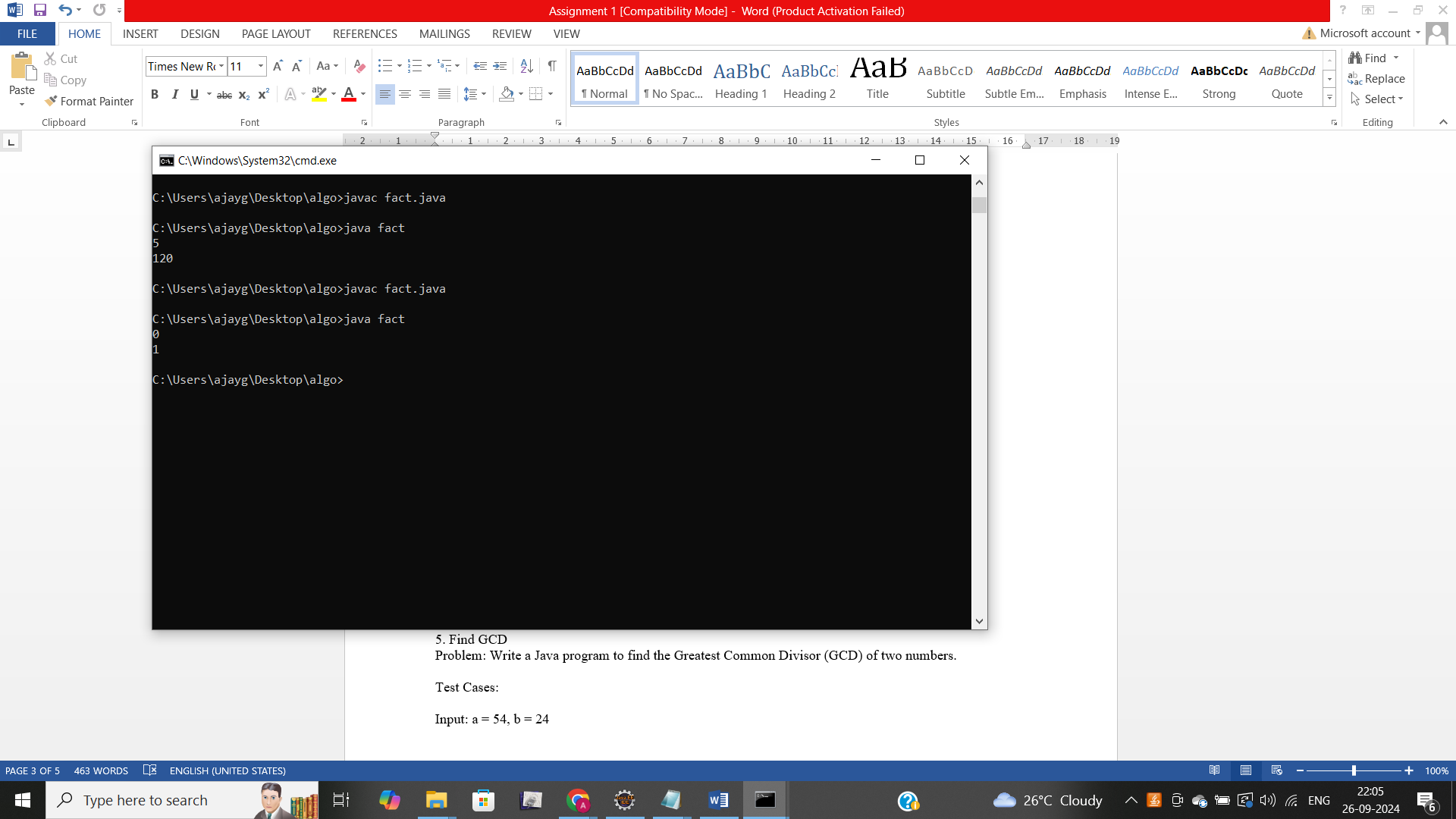
Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

System.out.println(factorial(num));

}

}



4. Fibonacci Series

Problem: Write a Java program to print the first n numbers in the Fibonacci series.

Test Cases:

Input: n = 5

Output: [0, 1, 1, 2, 3]

Input: n = 8

Output: [0, 1, 1, 2, 3, 5, 8, 13]

import java.util.Scanner;

class fibo{

static int fibonacii(int num){

if(num<=1){

return 1;

}

return fibonacii(num-1)+fibonacii(num-2);

}

public static void main(String [] args)

{

Scanner sc= new Scanner(System.in);

int num;

num=sc.nextInt();

System.out.print("[");

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

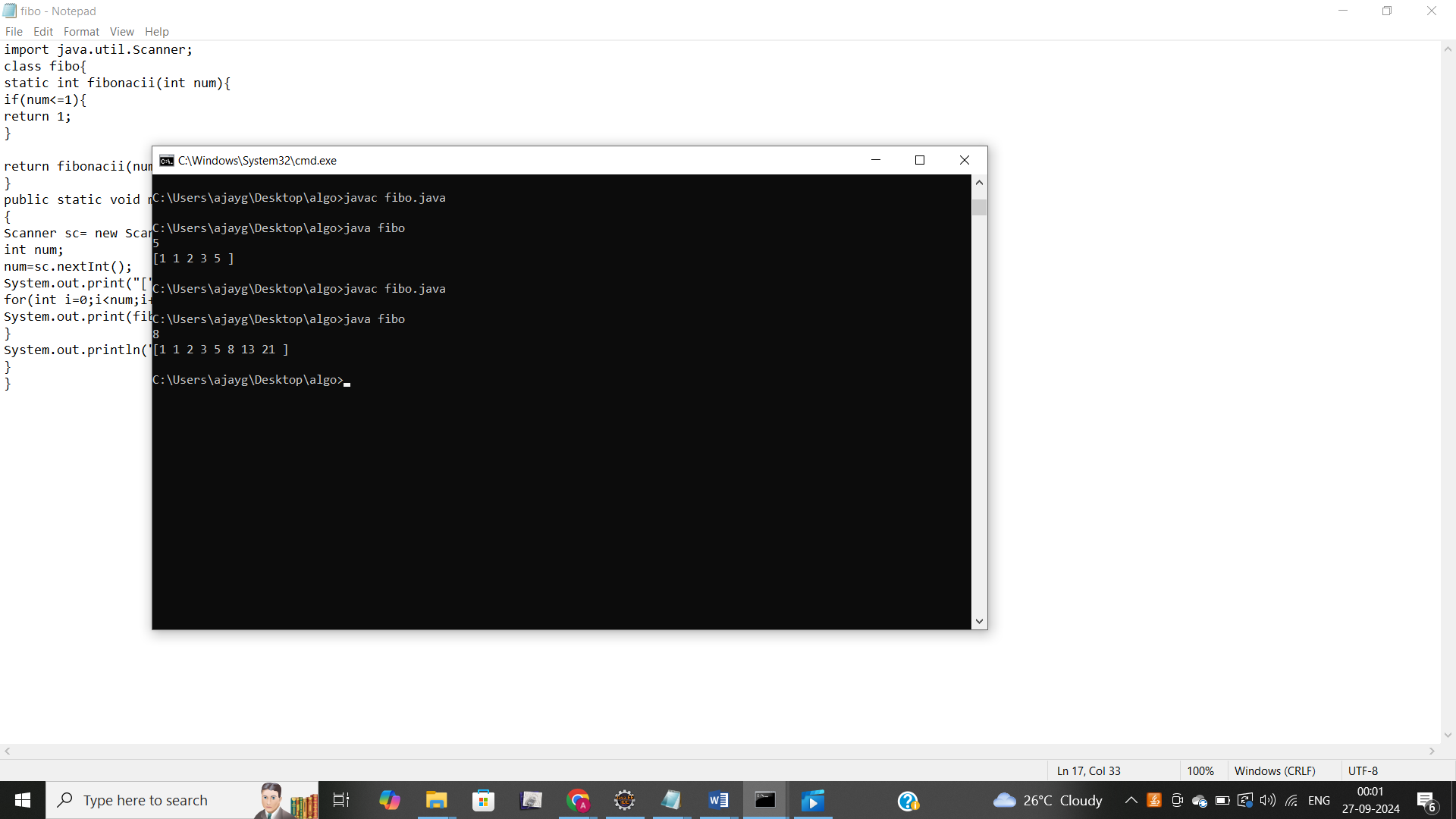
System.out.print(fibonacii(i)+" ");

}

System.out.println("]");

}

}



5. Find GCD

Problem: Write a Java program to find the Greatest Common Divisor (GCD) of two numbers.

Test Cases:

Input: a = 54, b = 24

Output: 6

Input: a = 17, b = 13

Output: 1

6. Find Square Root

Problem: Write a Java program to find the square root of a given number (using integer approximation).

Test Cases:

Input: x = 16

Output: 4

Input: x = 27

Output: 5

import java.util.Scanner;

class sqr{

static int squareroot(int x,int low,int high){

if(low>high){

return high;

}

int mid=(low+high)/2;

if(mid\*mid==x){

return mid;

}

if (mid\*mid>x){

return squareroot(x,low,mid-1);

}

return squareroot(x,mid+1,high);

}

public static void main(String []args){

Scanner sc=new Scanner(System.in);

int x;

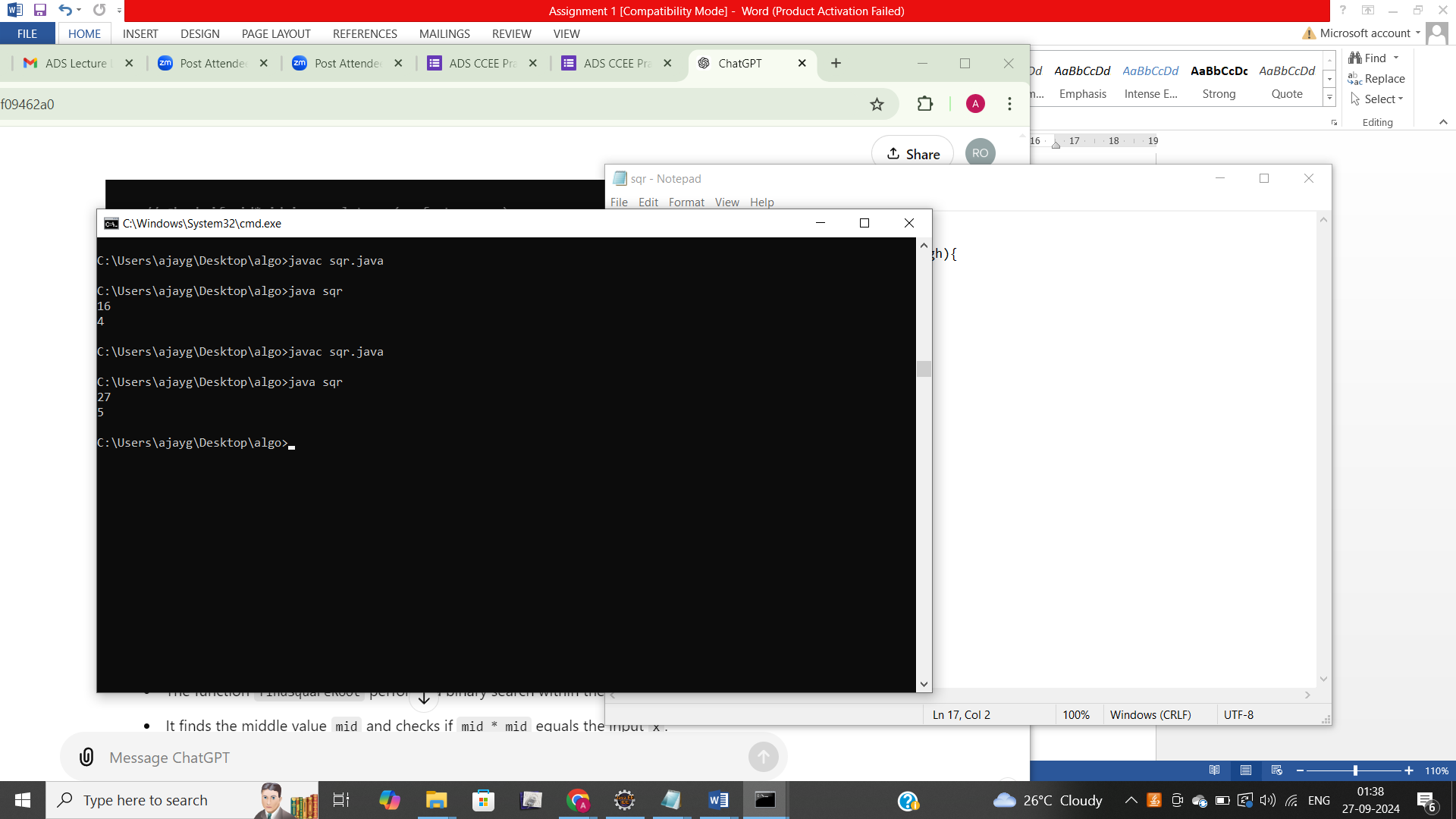
x=sc.nextInt();

int result = squareroot(x,0,x);

System.out.println(result);

}

}



7. Find Repeated Characters in a String

Problem: Write a Java program to find all repeated characters in a string.

Test Cases:

Input: "programming"

Output: ['r', 'g', 'm']

Input: "hello"

Output: ['l']

import java.util.\*;

public class repeted {

static void repetedchar(String str){

int len=str.length();

char[] chars=str.toCharArray();

Arrays.sort(chars);

String sorted=new String(chars);

System.out.print("[");

for(int i=0;i<len;i++) {

int count=1;

while(i < len-1&&sorted.charAt(i)==sorted.charAt(i+1)) {

count++;

i++;

}

if(count>1) {

System.out.print(sorted.charAt(i)+" ");

}

}

System.out.print("]");

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

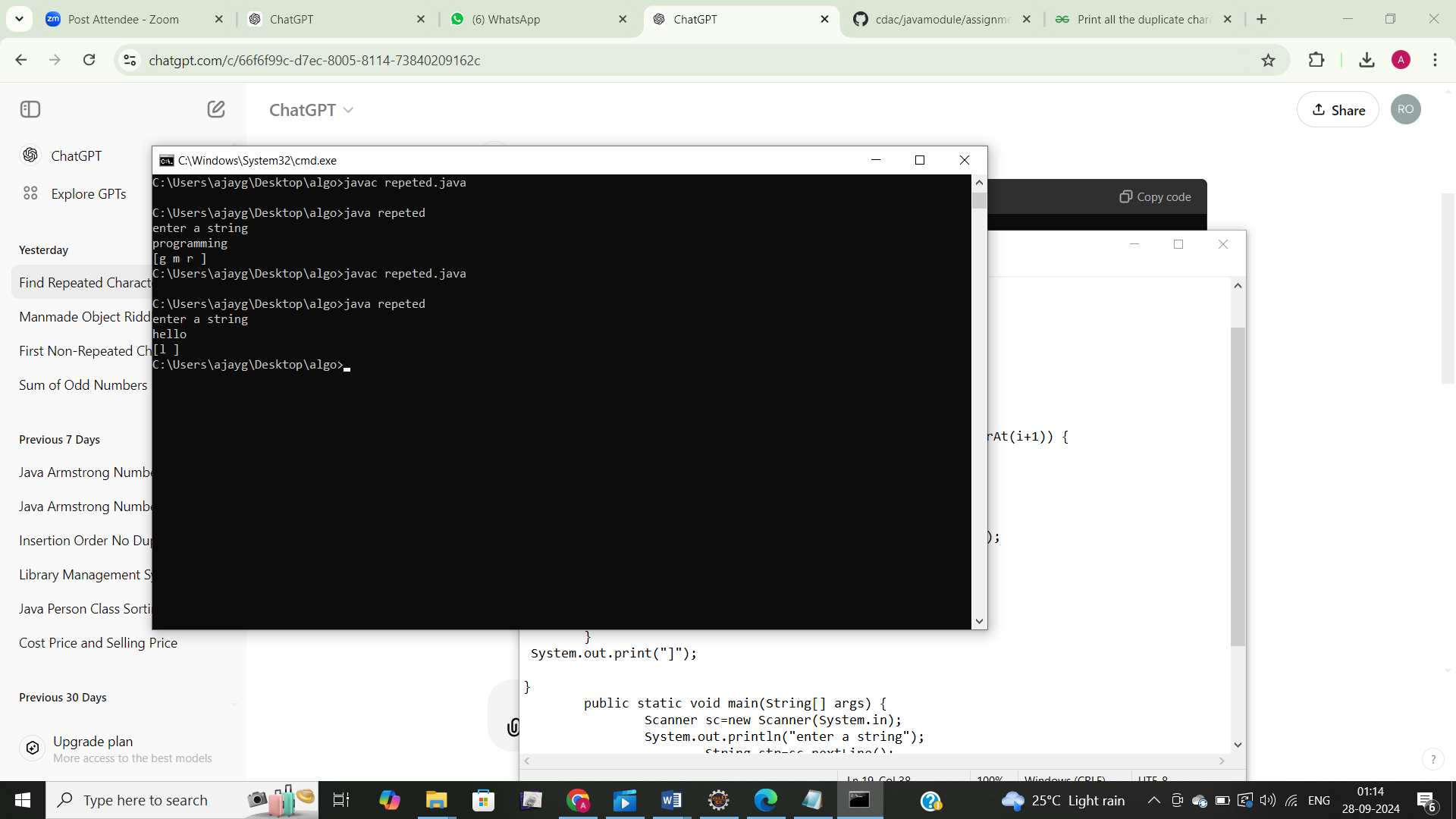
System.out.println("enter a string");

String str=sc.nextLine();

repetedchar(str);

}

}



8. First Non-Repeated Character

Problem: Write a Java program to find the first non-repeated character in a string.

Test Cases:

Input: "stress"

Output: 't'

Input: "aabbcc"

Output: null

**import** java.util.Scanner;

**import** java.util.Scanner;

**public** **class** first {

//String str;

//int index;

**static** Character firstnonrepeted(String str, **int** index) {

**if**(index>=str.length()) {

**return** **null**;

}

**char** currentchar=str.charAt(index);

**if**(str.indexOf(currentchar)==str.lastIndexOf(currentchar))

{

**return** currentchar;

}

**return** *firstnonrepeted*(str,index + 1);

}

//public static Character firstnonrepeted(String str) {

//return firstnonrepeted(str,0);

// }

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

String str= "stress";//sc.nextLine();

System.***out***.println(*firstnonrepeted*(str,0));

String str1="aabbcc";

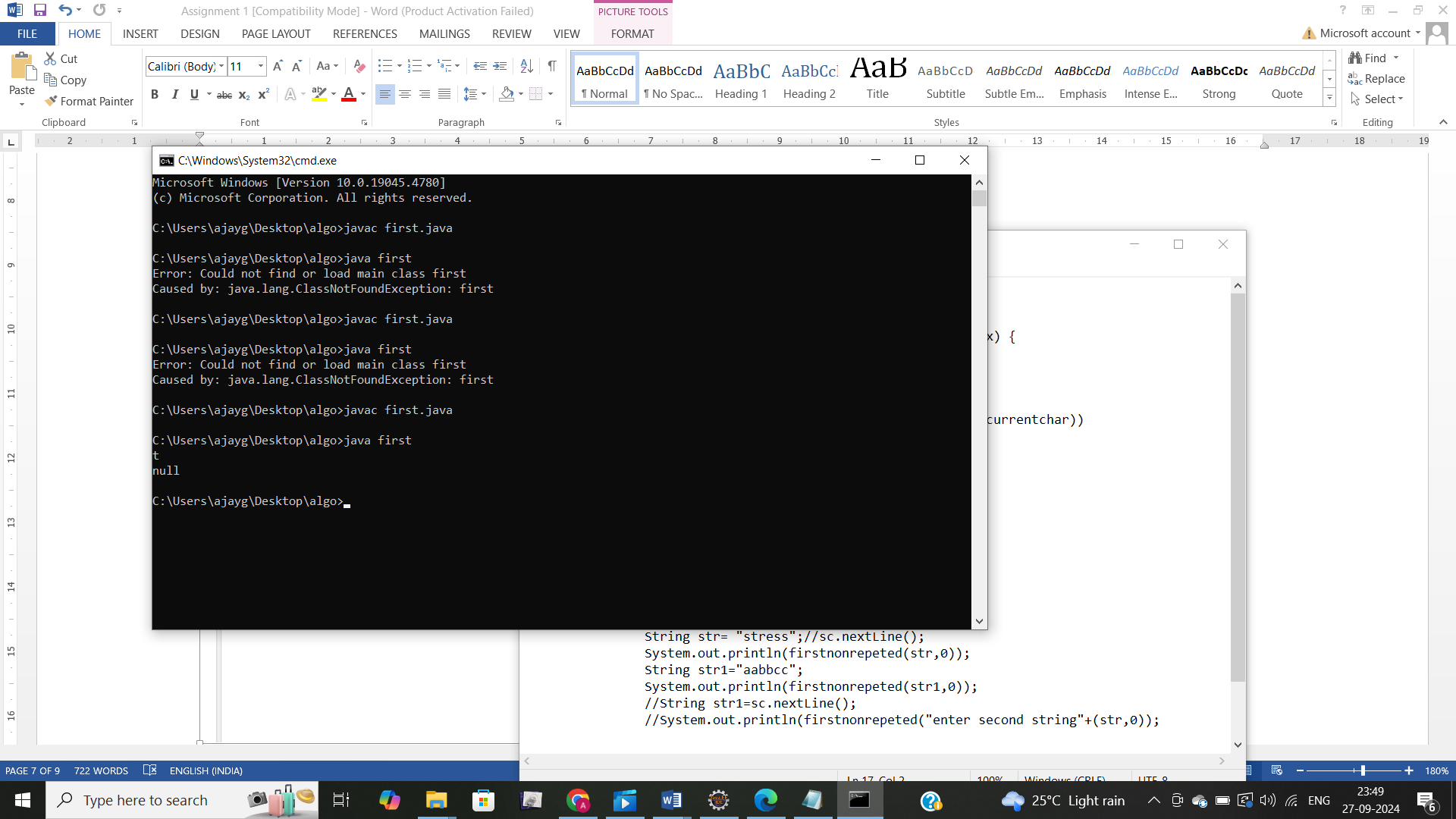
System.***out***.println(*firstnonrepeted*(str1,0));

//String str1=sc.nextLine();

//System.out.println(firstnonrepeted("enter second string"+(str,0));

}

}



9. Integer Palindrome

Problem: Write a Java program to check if a given integer is a palindrome.

Test Cases:

Input: 121

Output: true

Input: -121

Output: false

import java.util.Scanner;

class palin{

static boolean palindrome(int number){

if(number<0){

return false;

}

int originalnum=number;

int reverse=0;

while(number!=0){

int val=number%10;

reverse=reverse\*10+val;

number/=10;

}

return originalnum==reverse;

}

public static void main(String [] args){

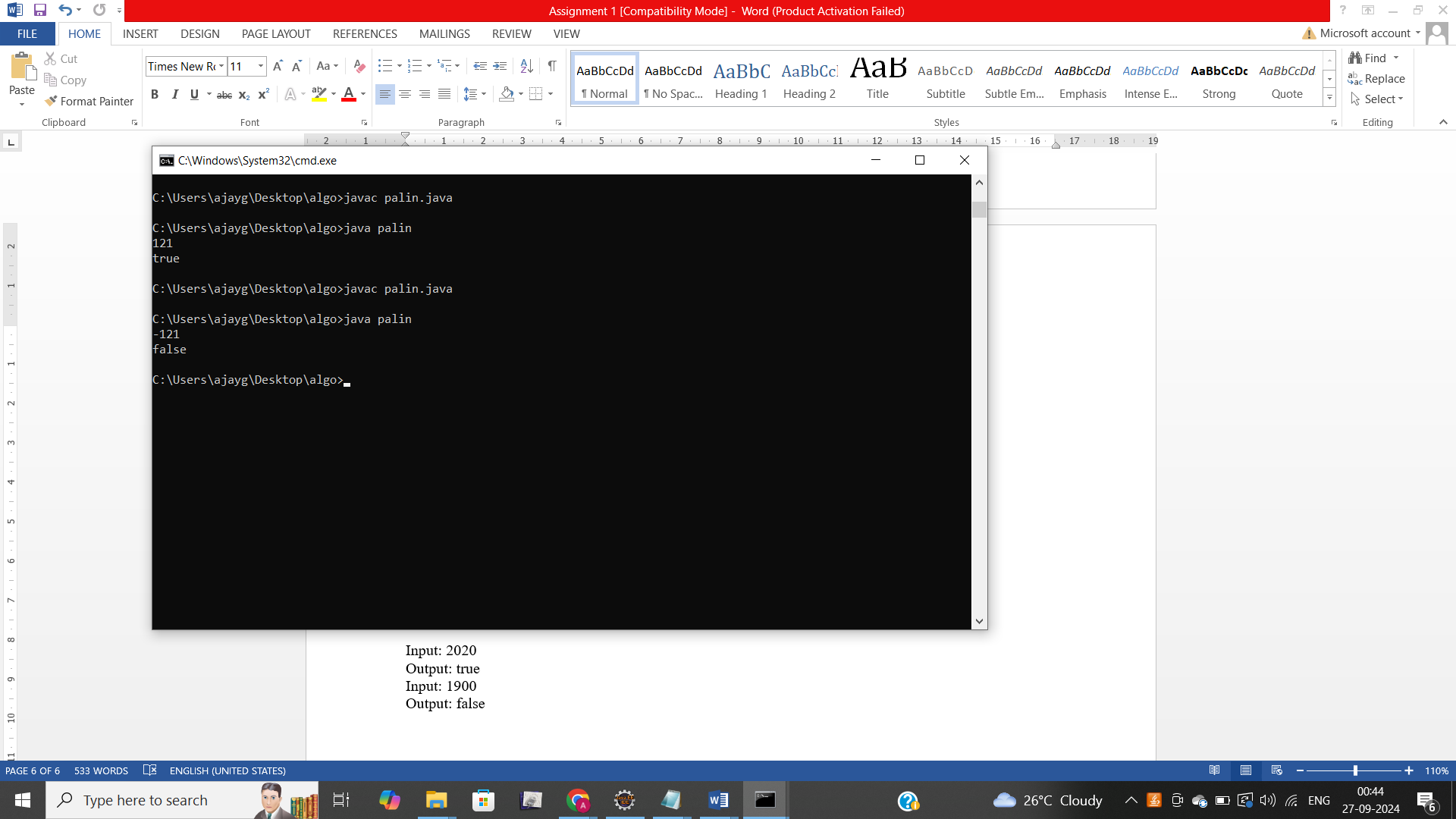
Scanner sc=new Scanner(System.in);

int number=sc.nextInt();

System.out.println(palindrome(number));

}

}



10. Leap Year

Problem: Write a Java program to check if a given year is a leap year.

Test Cases:

Input: 2020

Output: true

Input: 1900

Output: false

import java.util.Scanner;

class leapyear{

static boolean isLeaporNot(int year){

if(year%4==0){

if(year%100==0){

return year%400==0;

}

else{

return true;

}

}

return false;

}

public static void main(String [] args){

Scanner sc=new Scanner(System.in);

int year=sc.nextInt();

System.out.println(isLeaporNot(year));

}

}

