**C-DAC Mumbai Date 26/09/2024**

**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. Printing Patterns

Problem: Write a Java program to print patterns such as a right triangle of stars.

Test Cases:

Input: n = 3

Output:

\*

\*\*

\*\*\*

Input: n = 5

Output:

\*

\*\*

\*\*\*

import java.util.Scanner;

public class rightanglestar {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.print("enter number");

int size=sc.nextInt();

for(int i=1;i<=size;i++) {

for(int j=1;j<=i;j++) {

System.out.print("\*");

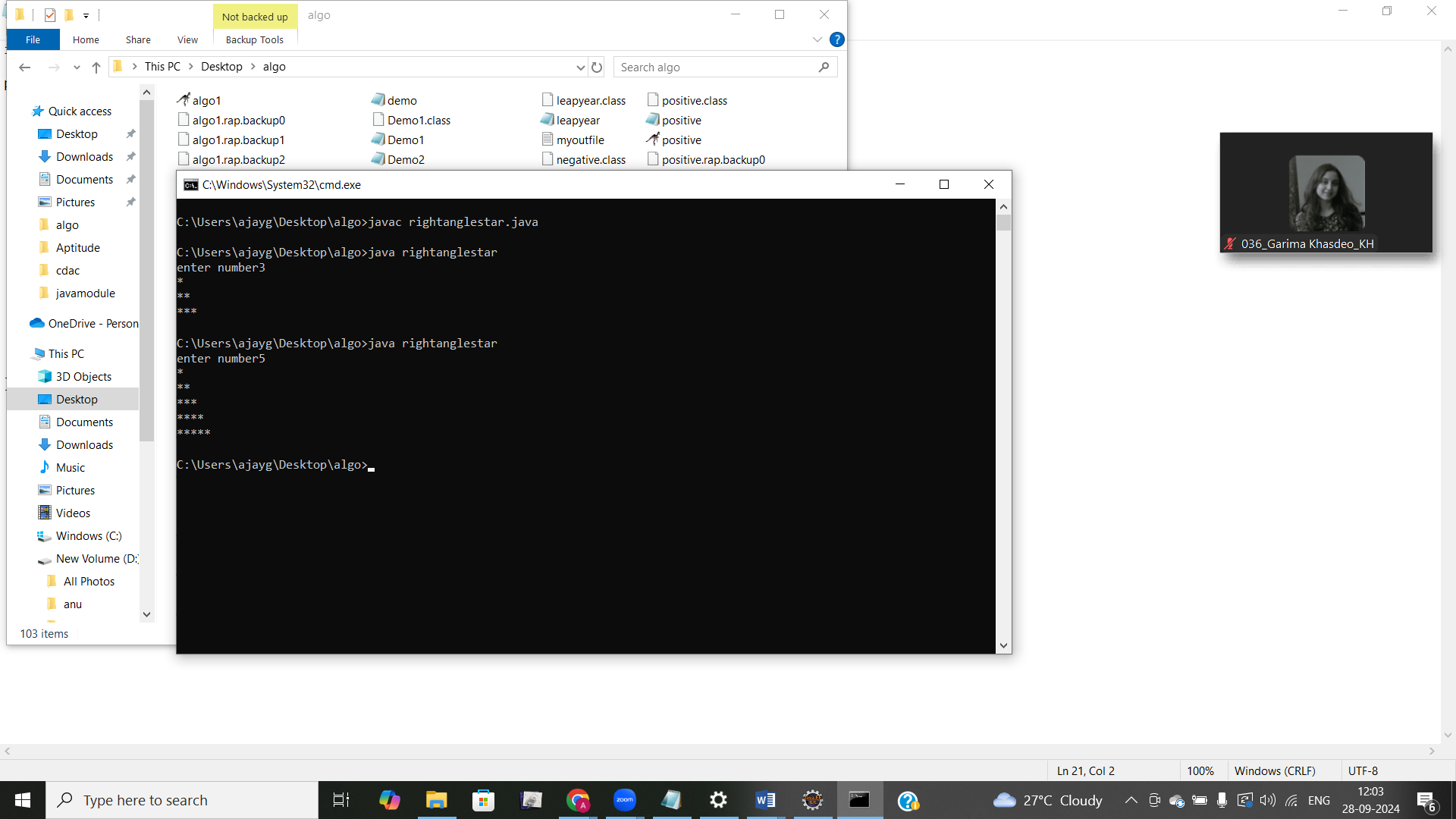
}

System.out.println();

}

}

}



2. Remove Array Duplicates

Problem: Write a Java program to remove duplicates from a sorted array and return the new length of the array.

Test Cases:

Input: arr = [1, 1, 2]

Output: 2

Input: arr = [0, 0, 1, 1, 2, 2, 3, 3]

Output: 4

import java.util.Scanner;

class removeduplicate{

static int removesame(int []arr)

{

if(arr.length==0) {

return 0;

}

int uniqueindex =1;

for(int i=1;i<arr.length;i++) {

if(arr[i]!=arr[i-1]) {

arr[uniqueindex++]=arr[i];

}

}

return uniqueindex;

}

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.println("enter size");

int size=sc.nextInt();

int [] arr=new int[size];

System.out.println("enter elments :");

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

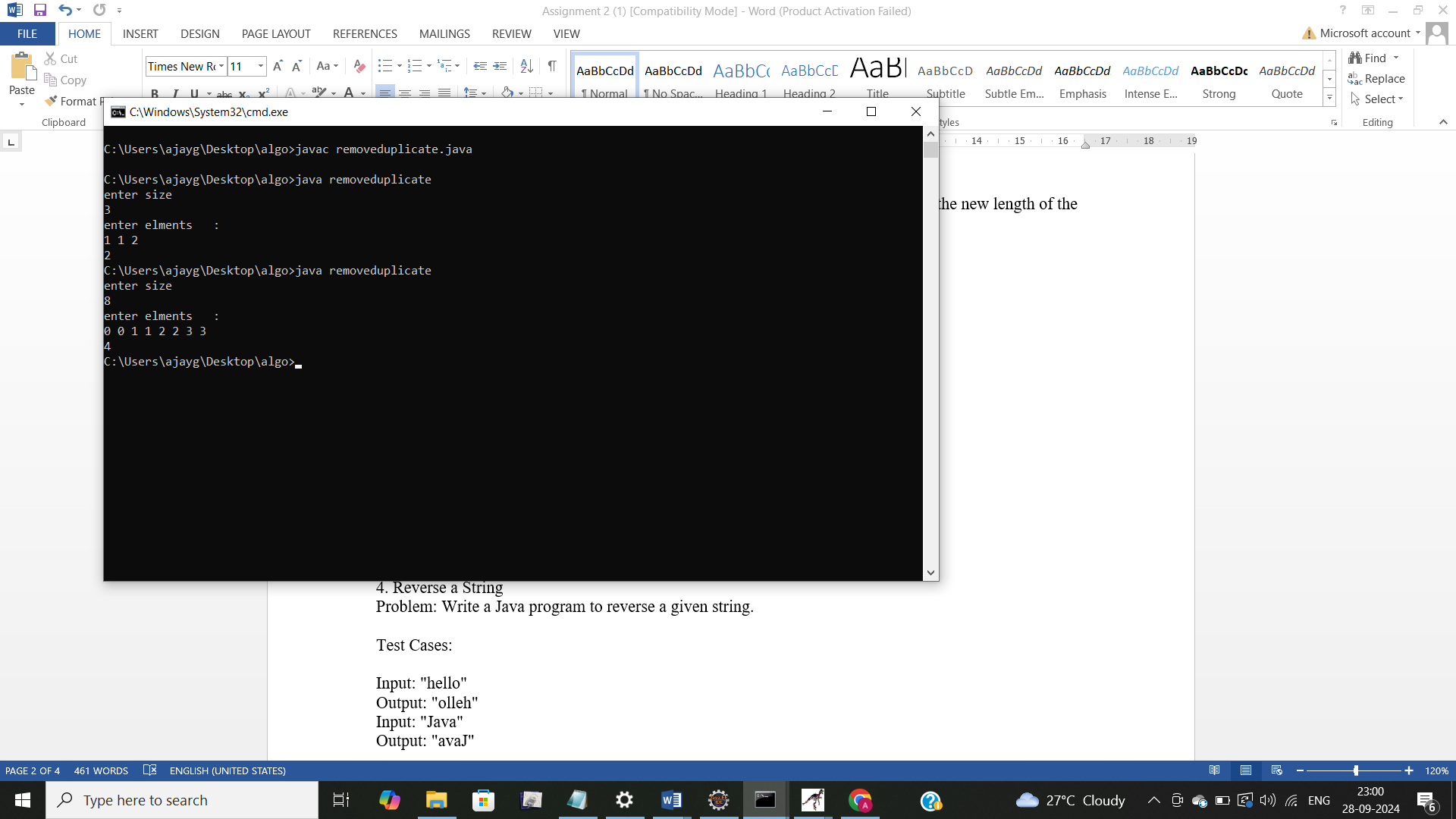
arr[i]=sc.nextInt();

}

System.out.print(removesame(arr));

}

}



3. Remove White Spaces from String

Problem: Write a Java program to remove all white spaces from a given string.

Test Cases:

Input: "Hello World"

Output: "HelloWorld"

Input: " Java Programming "

Output: "JavaProgramming"

import java.util.Scanner;

public class removespace {

public static String removespace1(String str) {

if(str.isEmpty()) {

return str;

}

if(str.charAt(0) == ' ') {

return removespace1(str.substring(1));

}

else {

return str.charAt(0) + removespace1(str.substring(1));

}

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

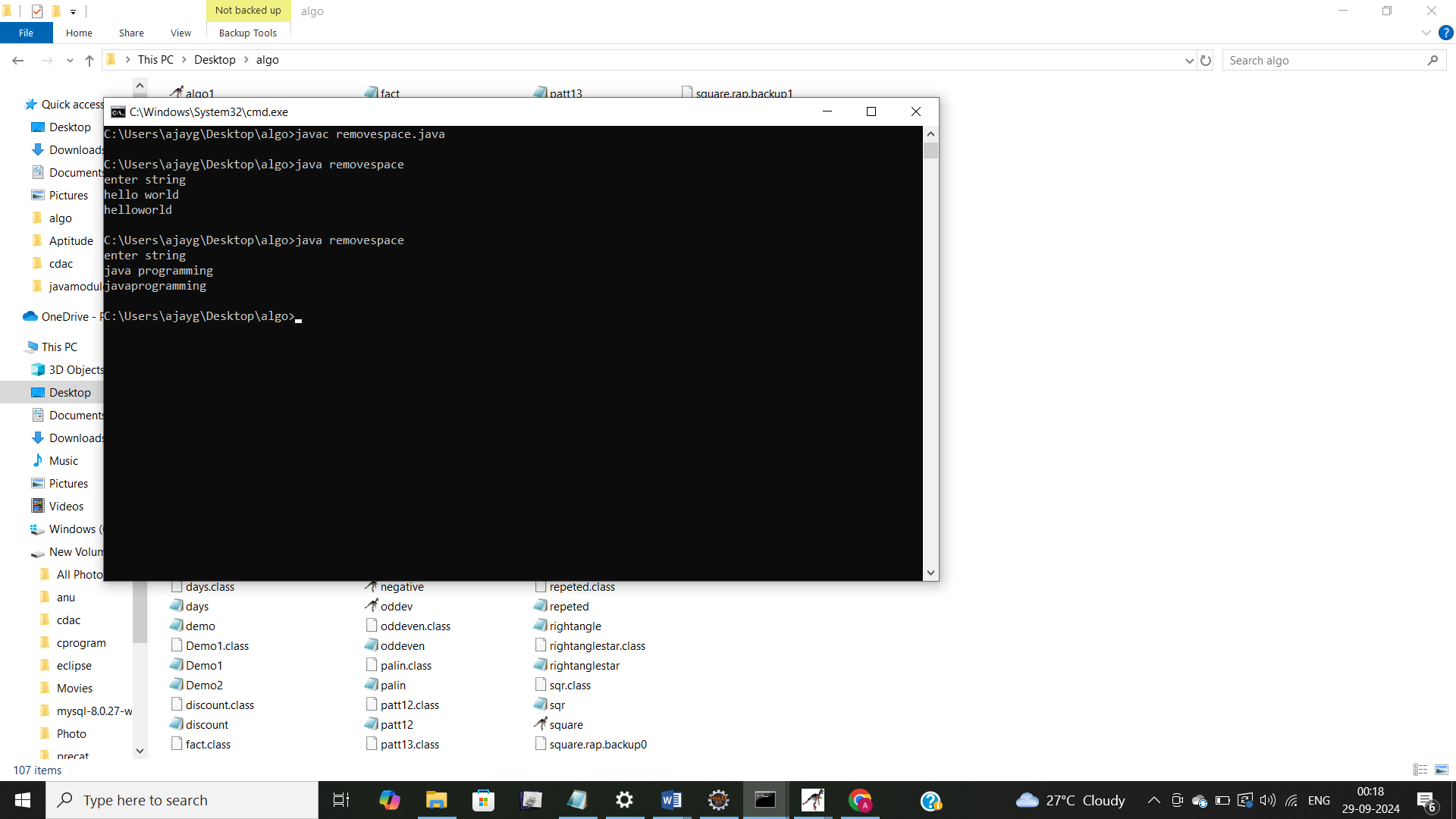
System.out.println("enter string");

String str=sc.nextLine();

System.out.println(removespace1(str));

}

}



4. Reverse a String

Problem: Write a Java program to reverse a given string.

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

import java.util.Scanner;

public class reversestr {

static String reversestring(String str) {

if(str.isEmpty()) {

return str;

}

return reversestring(str.substring(1)) + str.charAt(0);

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

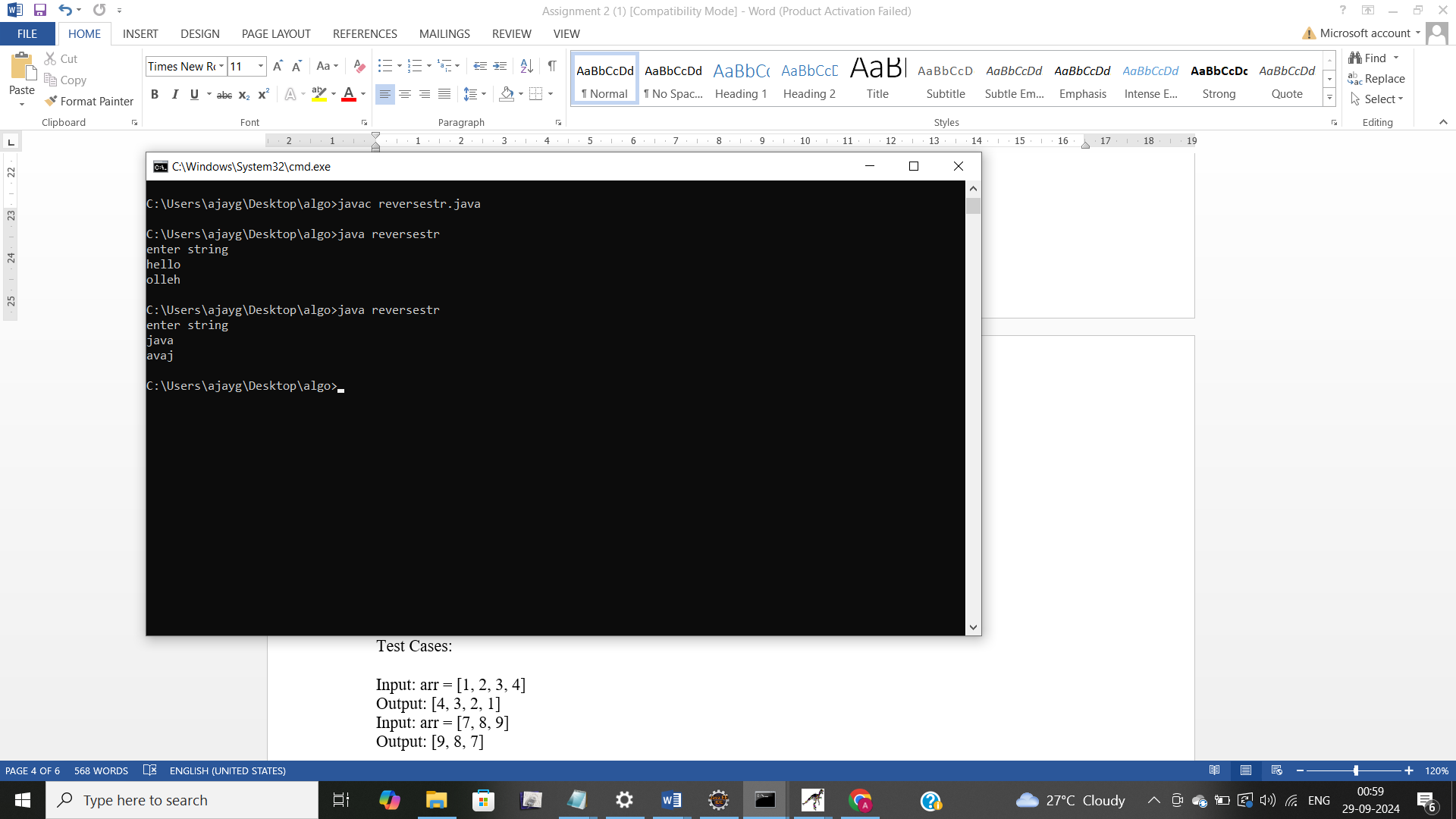
System.out.println("enter string");

String str= sc.nextLine();

System.out.println(reversestring(str));

}

}



5. Reverse Array in Place

Problem: Write a Java program to reverse an array in place.

Test Cases:

Input: arr = [1, 2, 3, 4]

Output: [4, 3, 2, 1]

Input: arr = [7, 8, 9]

Output: [9, 8, 7]

import java.util.Scanner;

public class revarray {

static void reverse(int[]arr,int n) {

if(n<=0) {

return;

}

System.out.print(arr[n-1]+" ");

reverse(arr,n-1);

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("enter size");

int n=sc.nextInt();

int []arr=new int[n];

System.out.println("enter elemnts");

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

{

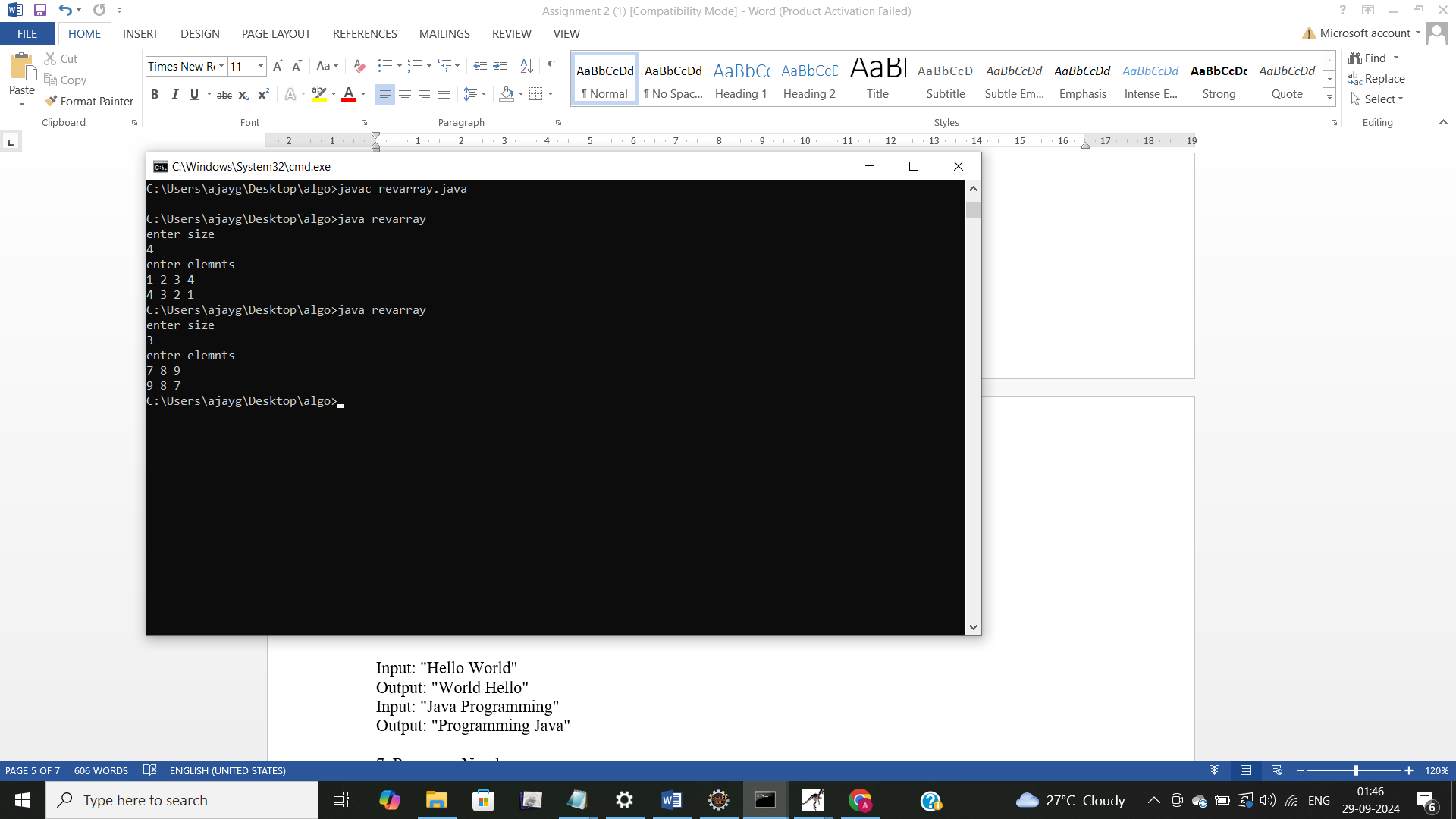
arr[i] =sc.nextInt();

}

reverse(arr,n);

}

}



6. Reverse Words in a String

Problem: Write a Java program to reverse the words in a given sentence.

Test Cases:

Input: "Hello World"

Output: "World Hello"

Input: "Java Programming"

Output: "Programming Java"

import java.util.Scanner;

public class revword {

static String reverseword(String sentense) {

if(!sentense.contains(" ")){

return sentense;

}

int spaceindex = sentense.indexOf(" ");

String firstword=sentense.substring(0,spaceindex);

String secondword=sentense.substring(spaceindex + 1);

String revremaining=reverseword(secondword);

return revremaining+ " "+firstword;

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

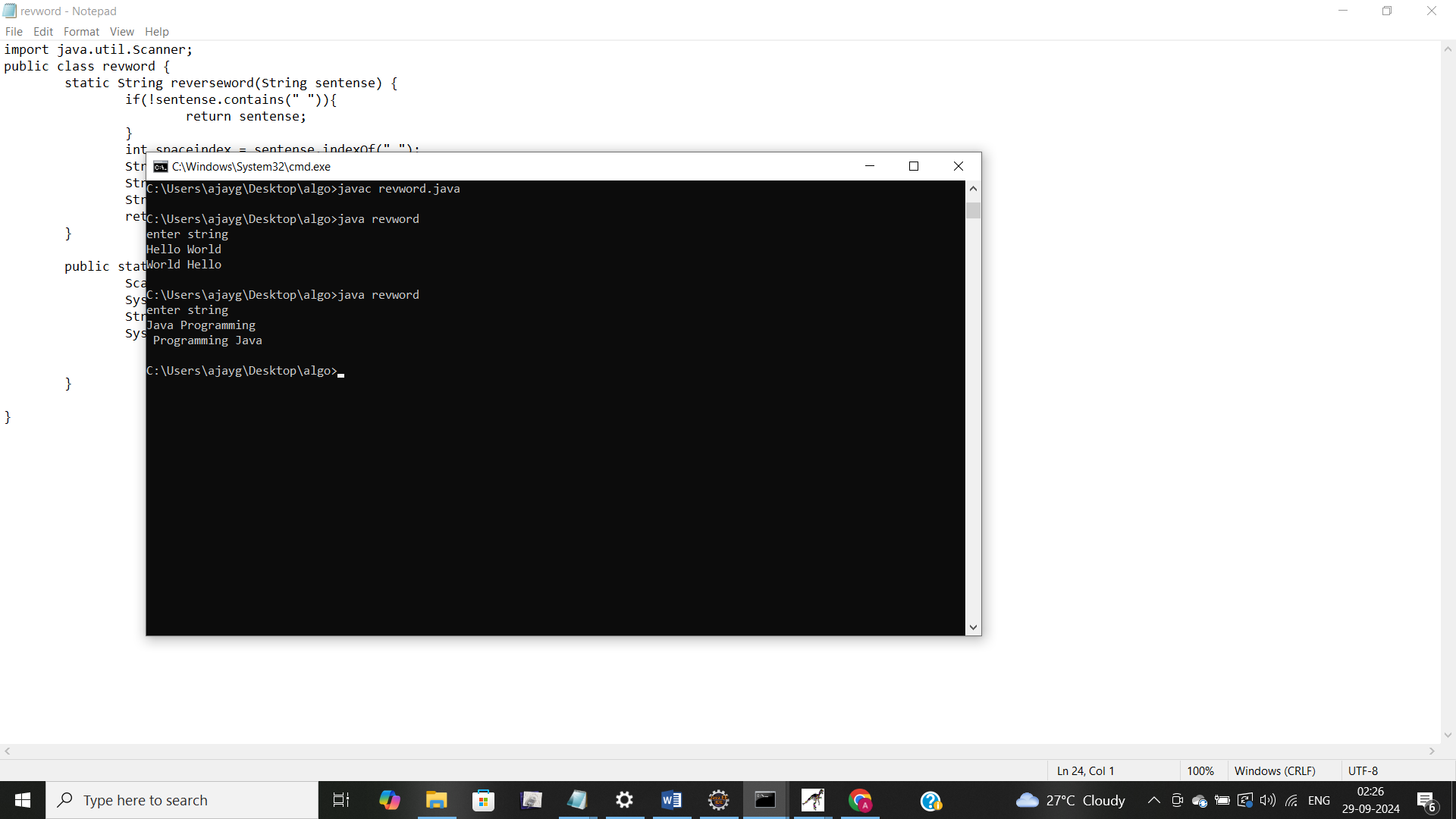
System.out.println("enter string");

String sentense=sc.nextLine();

System.out.println(reverseword(sentense));

}

}



7. Reverse a Number

Problem: Write a Java program to reverse a given number.

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

import java.util.Scanner;

public class revnum {

static int revformula(int num,int reverse) {

if(num==0) {

return reverse;

}

reverse=reverse\*10 +num %10;

return revformula(num/10, reverse);

}

static int reversenum(int num) {

int isnegative = num<0 ? -1:1;

num=Math.abs(num);

return revformula(num,0)\*isnegative;

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

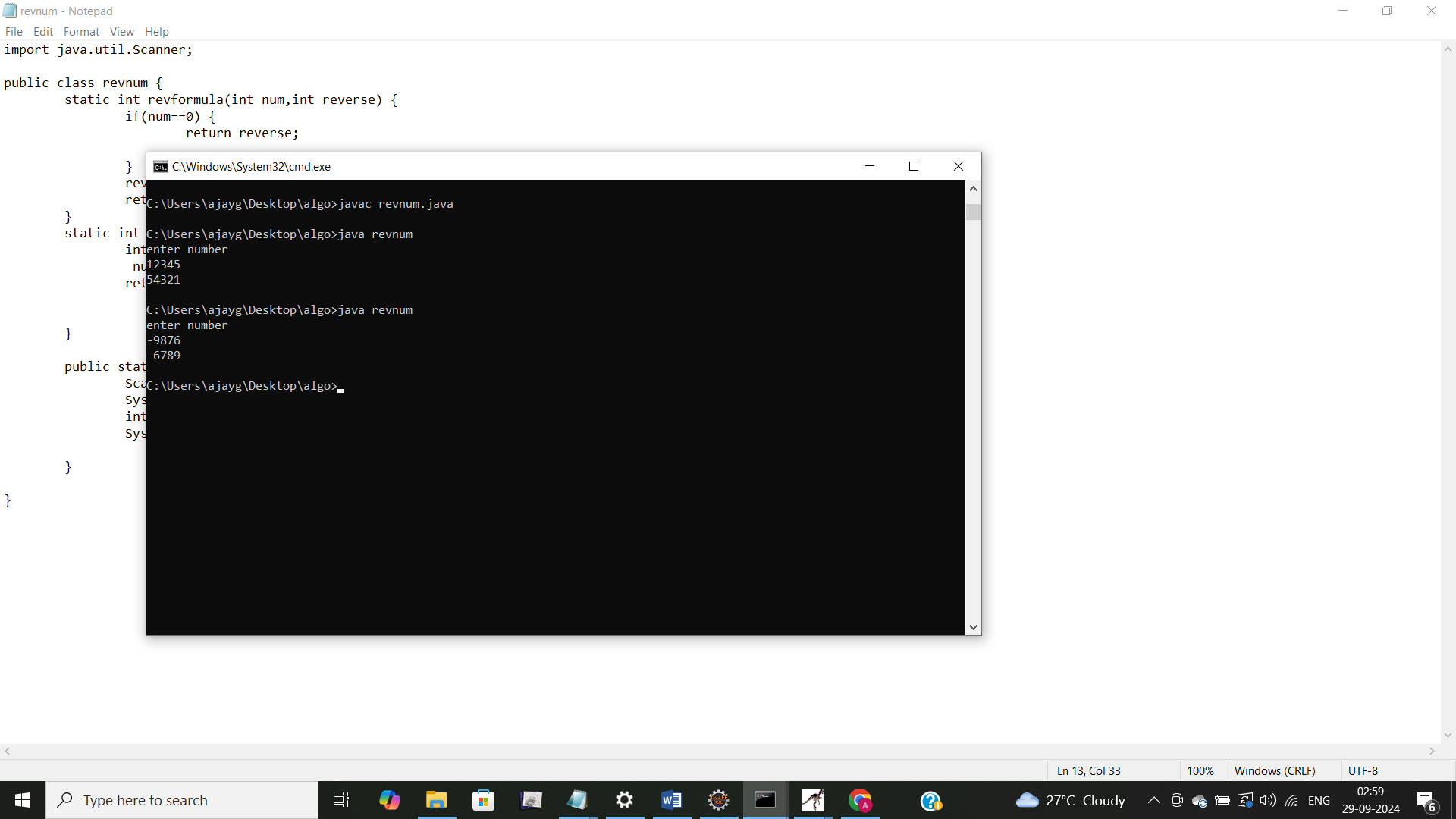
System.out.println("enter number");

int num=sc.nextInt();

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

}

}



8. Array Manipulation

Problem: Perform a series of operations to manipulate an array based on range update queries. Each query adds a value to a range of indices.

Test Cases:

Input: n = 5, queries = [[1, 2, 100], [2, 5, 100], [3, 4, 100]]

Output: 200

Input: n = 4, queries = [[1, 3, 50], [2, 4, 70]]

Output: 120

9. String Palindrome

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

Test Cases:

Input: "madam"

Output: true

Input: "hello"

Output: false

Here’s a continuation of the list of assignment questions starting from question 21, with two test cases for each:

**import** java.util.Scanner;

**public** **class** stringpali {

**public** **static** **boolean** palindrome(String str) {

**int** n=str.length();

**for**(**int** i=0;i<n/2;i++) {

**while**(str.charAt(i)!=str.charAt(n-i-1)) {

**return** **false**;

}

}

**return** **true**;

}

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

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

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

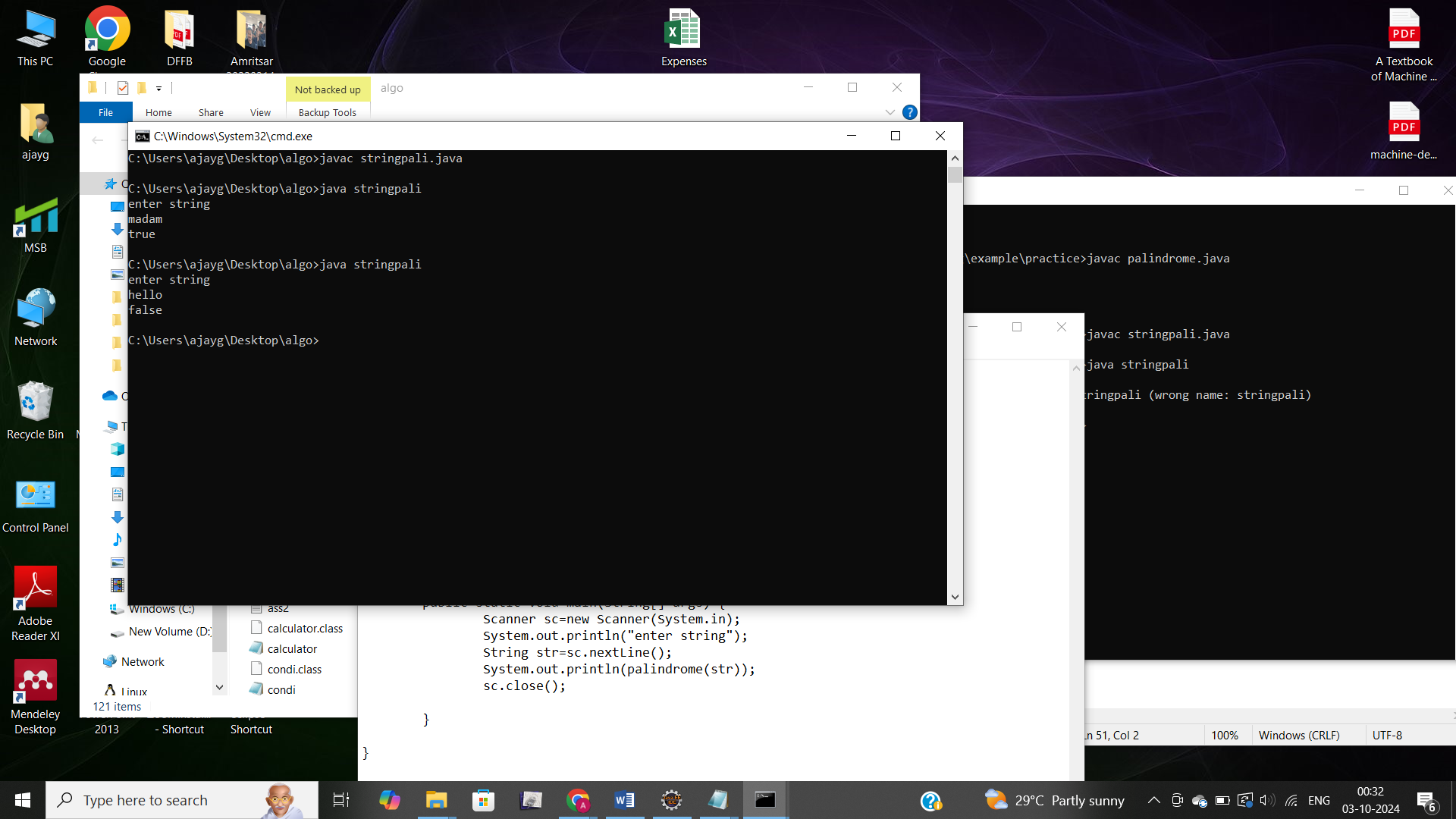
String str=sc.nextLine();

System.***out***.println(*palindrome*(str));

sc.close();

}

}



10. Array Left Rotation

Problem: Write a Java program to rotate an array to the left by d positions.

Test Cases:

Input: arr = [1, 2, 3, 4, 5], d = 2

Output: [3, 4, 5, 1, 2]

Input: arr = [10, 20, 30, 40], d = 1

Output: [20, 30, 40, 10]

import java.util.Scanner;

import java.util.Arrays;

class LeftRotationWithInput {

public static void rotateLeft(int[] arr, int d) {

int n = arr.length;

d = d % n;

reverseArray(arr, 0, d - 1);

reverseArray(arr, d, n - 1);

reverseArray(arr, 0, n - 1);

}

public static void reverseArray(int[] arr, int start, int end) {

while (start < end) {

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int n = scanner.nextInt();

int[] arr = new int[n];

System.out.println("Enter the elements of the array:");

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

arr[i] = scanner.nextInt();

}

System.out.print("Enter the number of positions to rotate: ");

int d = scanner.nextInt();

rotateLeft(arr, d);

System.out.println("Rotated Array: " + Arrays.toString(arr));

scanner.close();

}

}

