**Assignment-3 MCA**

**NAME: RAMAVATH SANTHOSH**

**ROLL NO: 22MCF1R40**

**COURSE: OOPS(JAVA)**

**2ND SEMESTER, 1ST YEAR MCA**

1. Write a JAVA program to compute the following Series and print the value of ‘Y’.Y = 1 – (x2 /2!) + (x4 /4!) - (x6 /6!) + ………. till n terms.  
   Where, ‘x’ and ‘n’ is inputted by the user.

CODE:

import java.util.Scanner;

class seriesseq {

public static int fact(int n) {

int f = 1;

while (n > 0) {

f = f \* n;

n--;

}

return f;

}

public static void main(String args[]) {

double x, y = 1;

int n;

Scanner obj = new Scanner(System.in);

System.out.println("Enter value of x: ");

x = obj.nextDouble();

System.out.println("Enter value of n: ");

n = obj.nextInt();

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

int k = 2 \* i;

int m = fact(k);

if (i % 2 == 0) {

y = y + ((Math.pow(x, k)) / m);

} else {

y = y - ((Math.pow(x, k)) / m);

}

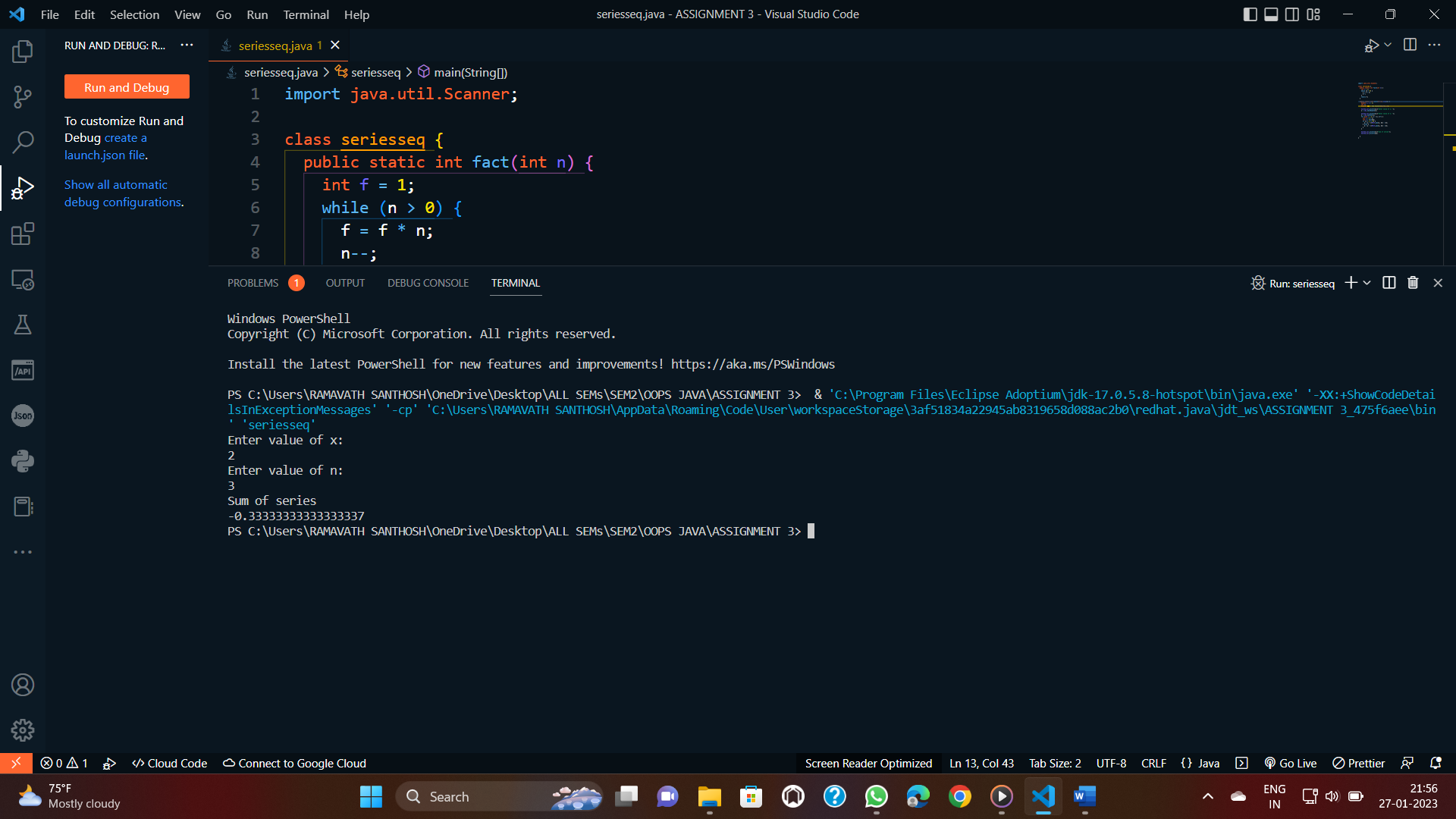
}

System.out.println("Sum of series");

System.out.println(y);

}

}



Wrtie a C function which takes string as input and reverse it, without using string  
predefined functions. e.g. “ABCDE” o/p: “EDCBA”

CODE:

import java.util.Scanner;

class strReverse {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

String s = sc.next();

System.out.println(" My Original String is : " + s);

System.out.println("The Reversed String is : " + reverse(s));

}

public static String reverse(String s) {

int i = 0, j = s.length() - 1;

char c[] = s.toCharArray();

while (i < j) {

char t = c[i];

c[i] = c[j];

c[j] = t;

i++;

j--;

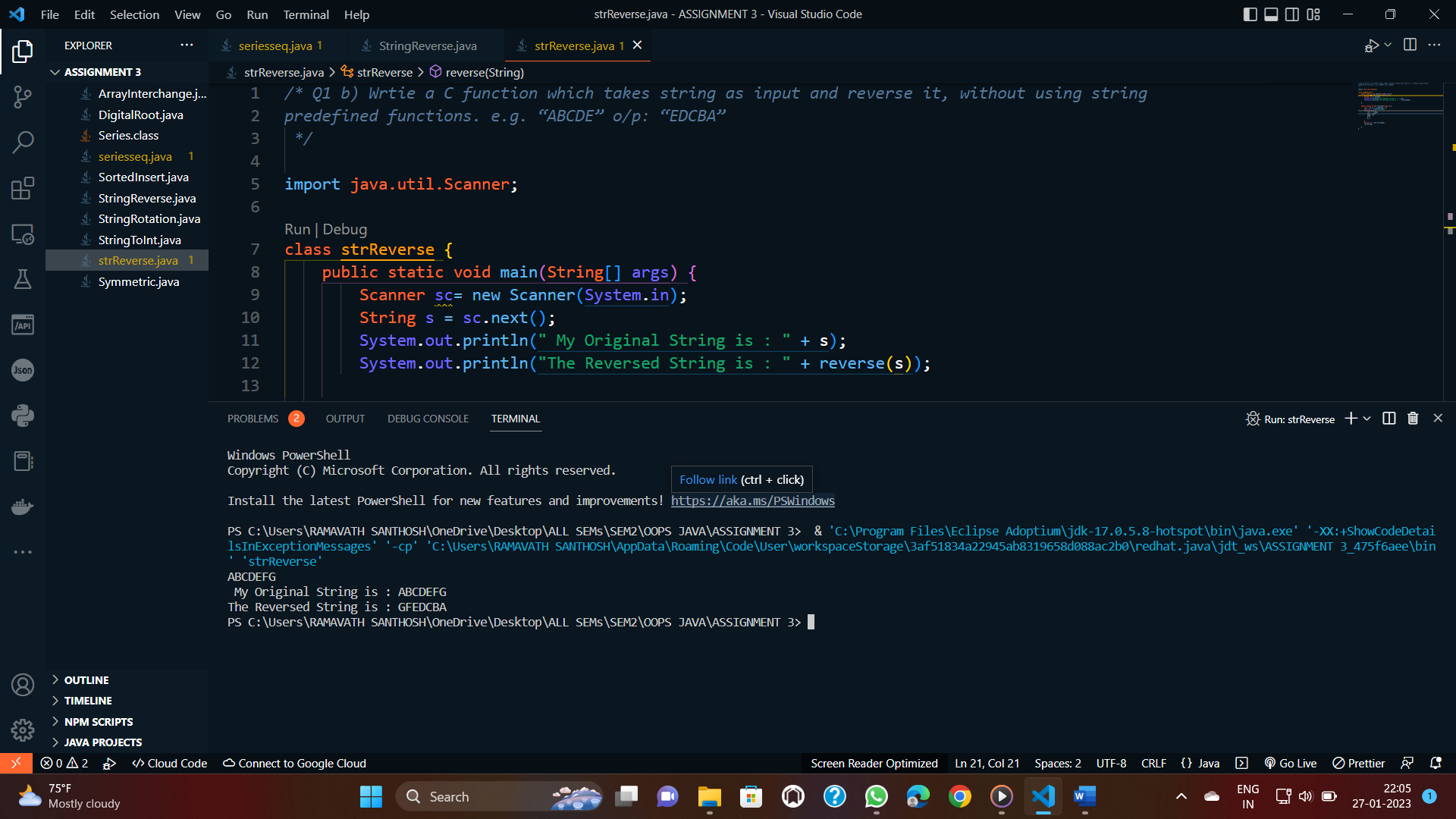
}

String str = new String(c);

return str;

}

}



1. Write a java method insert(), to insert an integer x into a sorted array A[] (sorted inascending order) containing N integers so that the array remains sorted after insertion. The  
   method takes x and array A as input and returns the index of x where it get inserted in A[].  
   Note: indicies starts from 0.

CODE:

import java.util.Arrays;

import java.util.Scanner;

public class SortedInsert {

    public static void main(String[] args) {

        int a[] = { 1, 2, 6, 7, 8 };

        System.out.println("Original Array: ");

        for (int i : a) {

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

        }

        System.out.println();

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a value to insert: ");

        int v = sc.nextInt();

        System.out.println("Element must be added at " + findPos(a, v));

        int[] newArray = new int[a.length + 1];

        int i = 0;

        int j = 0;

        while (a[i] < v && i < a.length) {

            newArray[j++] = a[i++];

        }

        newArray[j++] = v;

        while (i < a.length) {

            newArray[j++] = a[i++];

        }

        a = Arrays.copyOf(newArray, j - 1);

        System.out.println("New Array: ");

        for (int x : a) {

            System.out.print(x + " ");

        }

    }

    public static int findPos(int[] a, int value) {

        int i = 0;

        while (a[i] < value) {

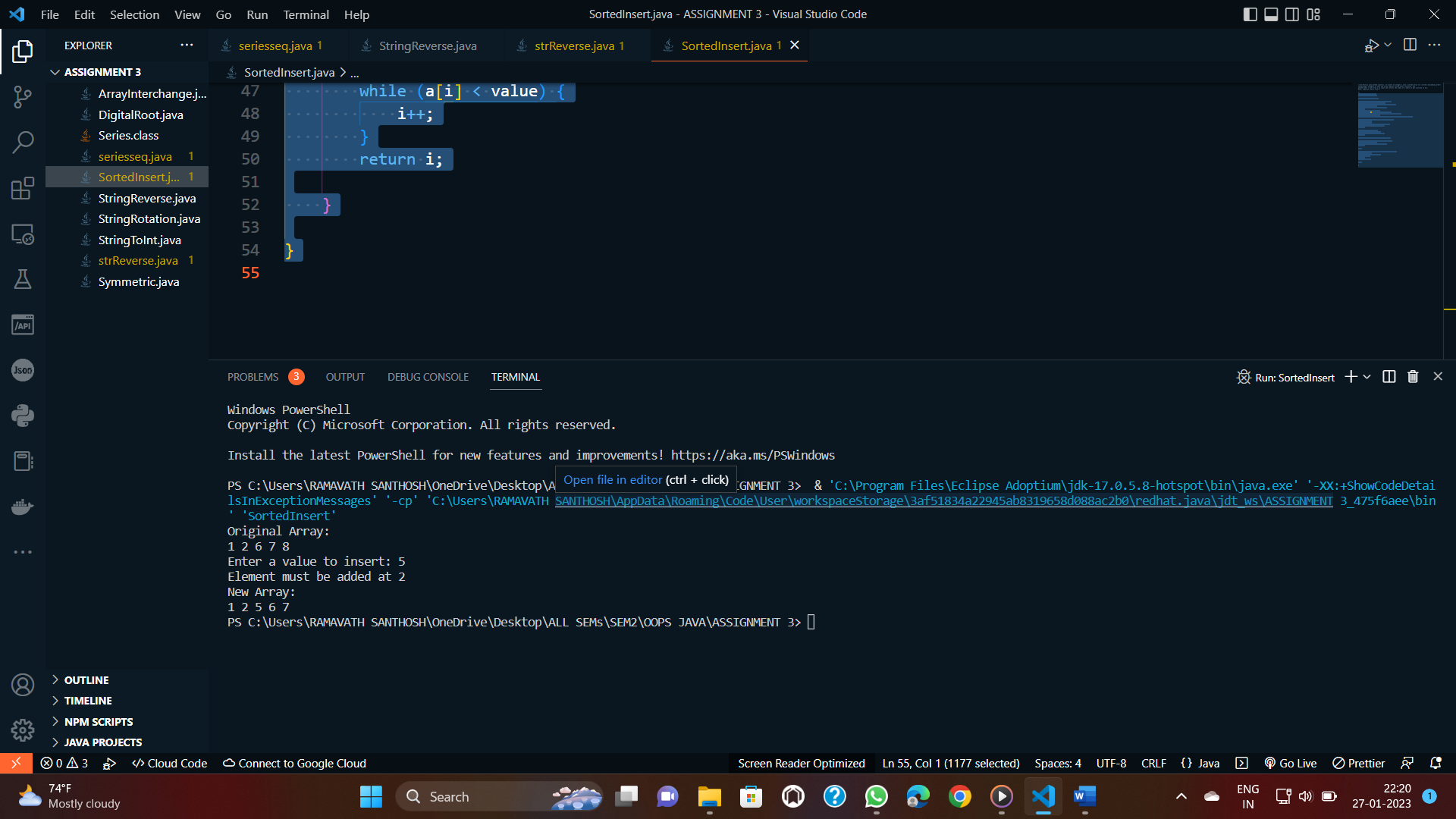
            i++;

        }

        return i;

    }

}



1. Write a java method which takes two dimensional integer array as input and prints whether the inputted matrix is symmetric.

Assume 2-D array is of size NxN, where N is known  
constant. A matrix is said to be symmetric if it equals to its transpose .

CODE:

import java.io.\*;

class Symmetric {

public static void main(String[] args)throws IOException {

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the dimension of matrix:");

int n=Integer.parseInt(br.readLine());

System.out.println("Enter the values of matrix rowwise: ");

int a[][]=new int[n][n];

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

{

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

{

a[i][j]=Integer.parseInt(br.readLine());

}

}

if(sym(a,n))

{

System.out.println("Matrix is Symmetric!!");

}

else

{

System.out.println("Matrix is not Symmetric!!");

}

}

public static boolean sym(int a[][],int n) {

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

{

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

{

if(a[i][j]!=a[j][i])

{

return false;

}

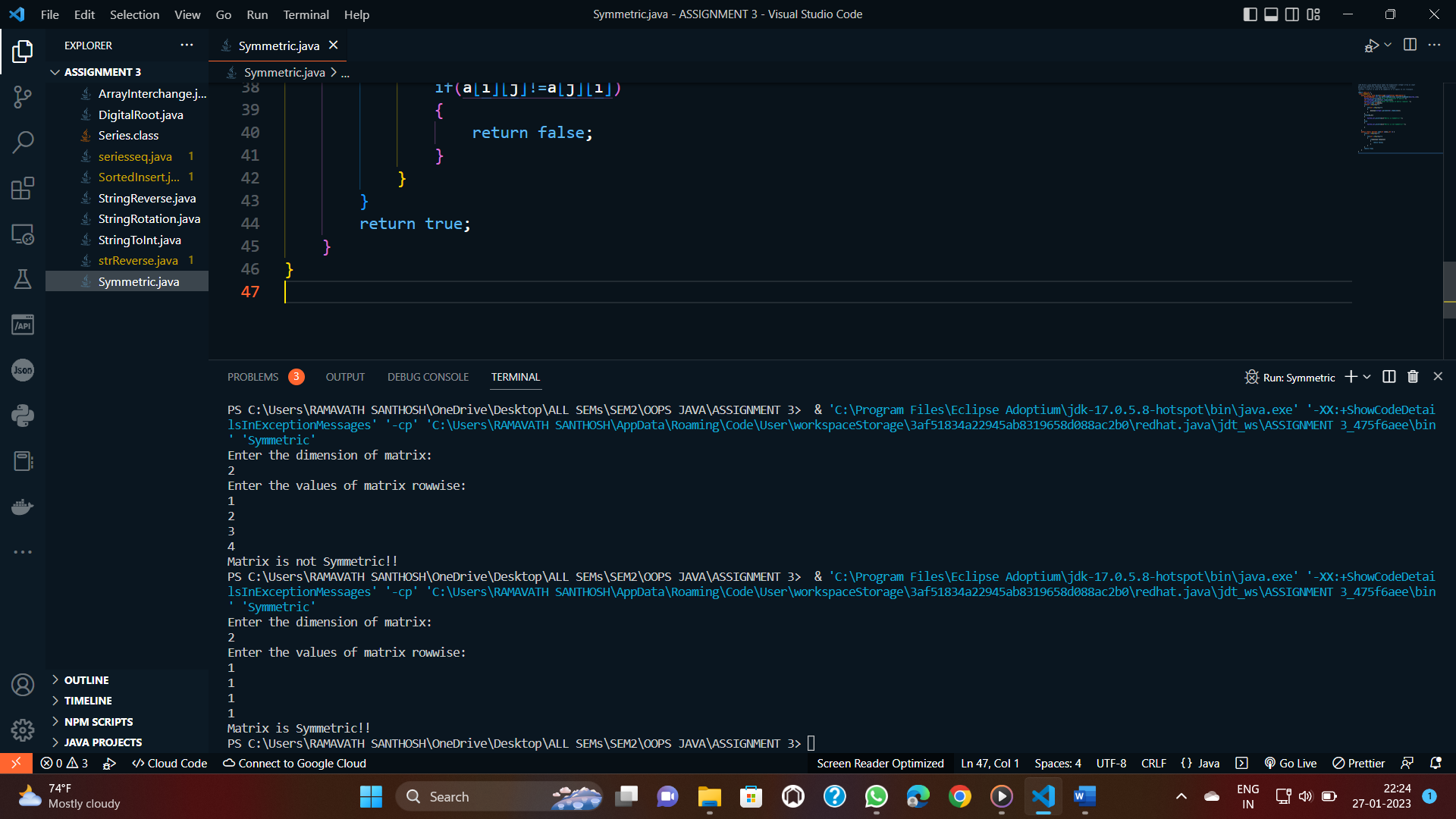
}

}

return true;

}

}



1. Write a JAVA program to extract numeric characters from a given string and display the integer sum of all the numeric character’s integral value.

CODE:

import java.io.\*;

class StringToInt {

    public static void main(String[] args)throws IOException {

        BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter numeric character: ");

        String s=(br.readLine());

        System.out.println(stoi(s));

    }

    public static int stoi(String s) {

        int i=0;

        int num=0;

        int sum=0;

        while(i<s.length())

        {

            char c=s.charAt(i);

            int x=(int)(c-'0');

            num=num\*10+x;

            sum+=x;

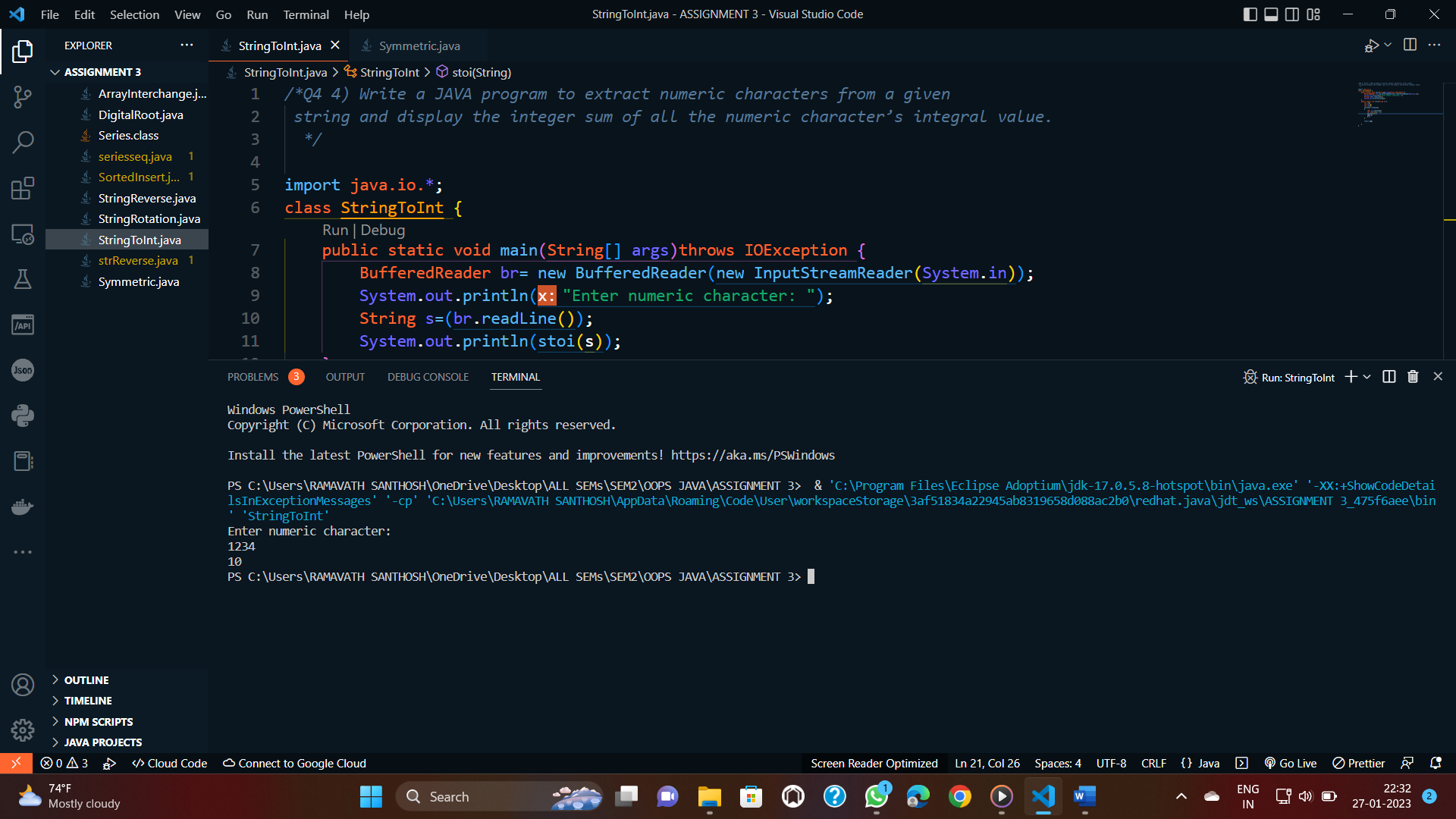
            i++;

        }

        return sum;

    }

}



1. Write a program to interchange corresponding elements of two single dimensional integer arrays of same size, say ‘n’. Use function to interchange the elements and pass arrays  
   as a parameters to the function.  
   E.g.: If A[]={1,2,3,4} and B[]={5,6,7,8} then A[]={5,6,7,8} and B[]={1,2,3,4}.

CODE:

import java.util.Arrays;

public class ArrayInterchange {

public static void main(String[] args) {

int[] A = {1, 2, 3, 4};

int[] B = {5, 6, 7, 8};

interchange(A, B);

System.out.println("My inputed string is A={1,2,3,4} and B= {5,6,7,8} ");

System.out.println("A: " + Arrays.toString(A));

System.out.println("B: " + Arrays.toString(B));

}

public static void interchange(int[] A, int[] B) {

for (int i = 0; i < A.length; i++) {

int temp = A[i];

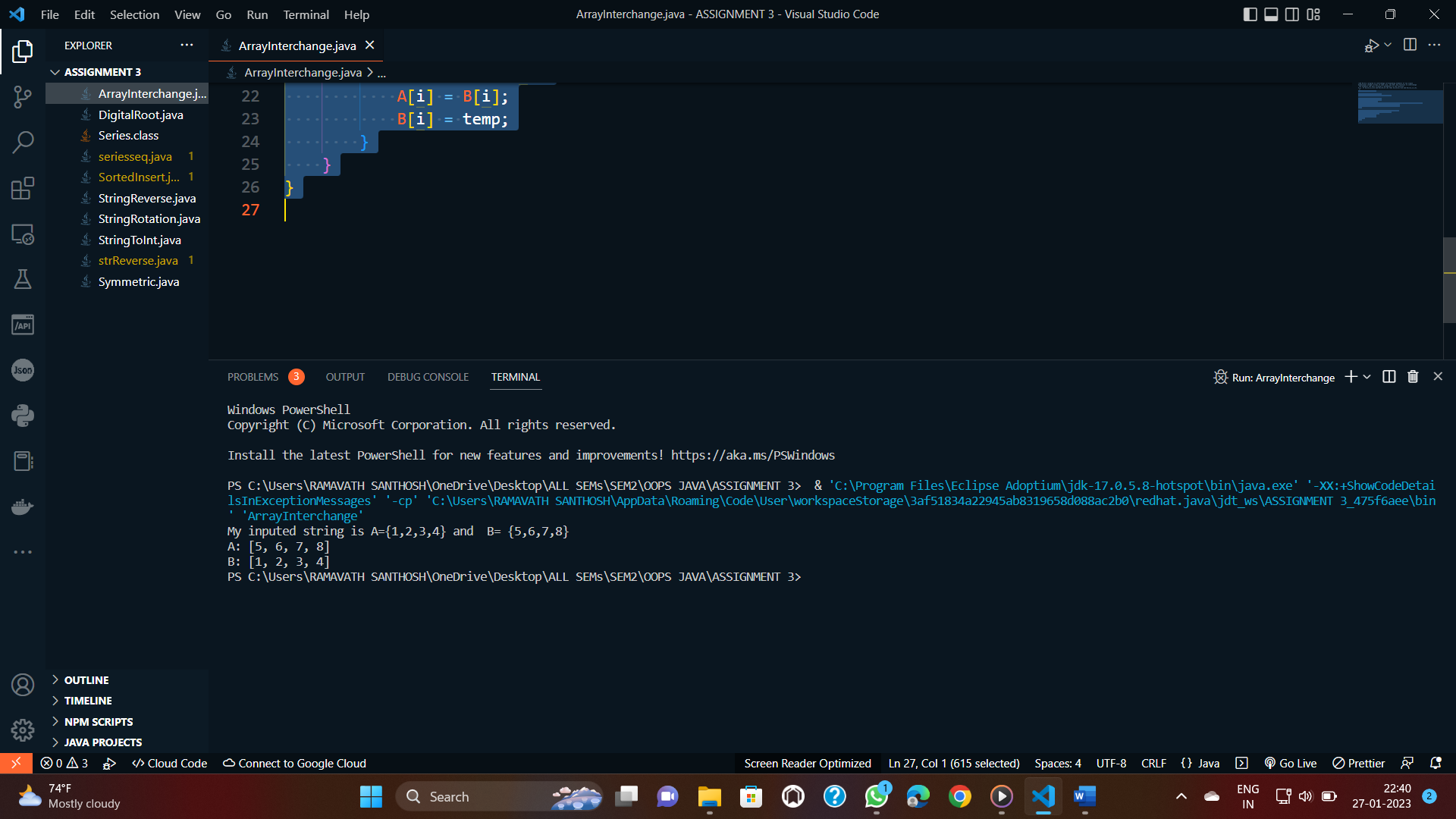
A[i] = B[i];

B[i] = temp;

}

}

}



1. Write a Method rotate( ) which takes an input string and integer d and shifts all characterin the input string towards right by d elements. For example if input string is “apple” and d=2  
   then the Method changes the string to “leapp”.

CODE:

import java.io.\*;

public class StringRotation {

public static void main(String[] args)throws IOException {

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the string: ");

String input=(br.readLine());

//String input = "apple";

int d= Integer.parseInt(br.readLine());

//int d = 2;

System.out.println(rotate(input, d));

}

public static String rotate(String input, int d) {

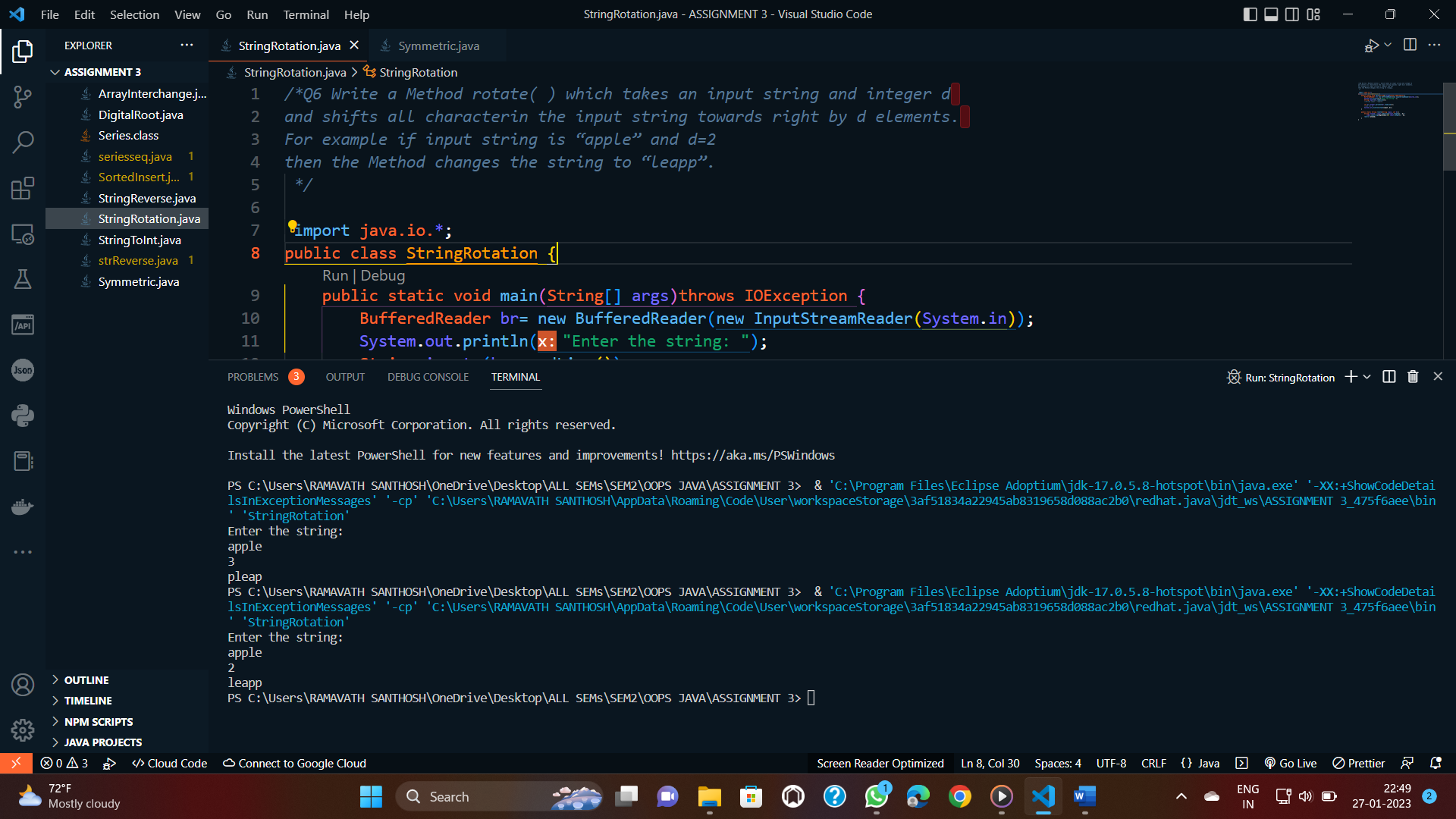
String rotated = input.substring(input.length() - d) +

input.substring(0, input.length() - d);

return rotated;

}

}



1. The digital root of an integer n is defined as the result of summing the digits repeatedly until only a single digit remains. For example, the digital root of 1729 can be calculated  
   using the following steps:  
   Step 1: 1+7+2+9 = 19  
   Step 2: 1+9 = 10  
   Step 3: 1+0 = 1  
   Because the total at the end of step 3 is the single digit 1, that value is the digital root.  
   Write a method digitalRoot(n) that returns the digital root of its argument. Write the  
   Method recursively without using any explicit loop constructs.

Note:- whatever concepts you know about PSCP use in java.

CODE:

import java.io.\*;

public class DigitalRoot {

public static void main(String[] args)throws IOException {

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter atleast a two digit number :");

int n=Integer.parseInt(br.readLine());

System.out.println(digitalRoot(n));

}

public static int digitalRoot(int n) {

if (n < 10) {

return n;

}

int sum = 0;

while (n > 0) {

sum += n % 10;

n /= 10;

}

return digitalRoot(sum);

}

}

