NAME : SOHAM SAHA

SEC : CS2A

ROLL : 61

ENROLLMENT : 12019009001389

SUB : OOP WITH JAVA LAB

Q1. Write a Java program to calculate Sum & Average of an integer array.

import java.util.\*;

public class sumavg {

public static void main(String args[]){

Scanner in = new Scanner (System.in);

System.out.print("Enter the number of elements : ");

int n = in.nextInt();

int s = 0;

int[] array = new int[100];

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

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

array[i] = in.nextInt();

s = s + array[i];

}

float avg = s/n;

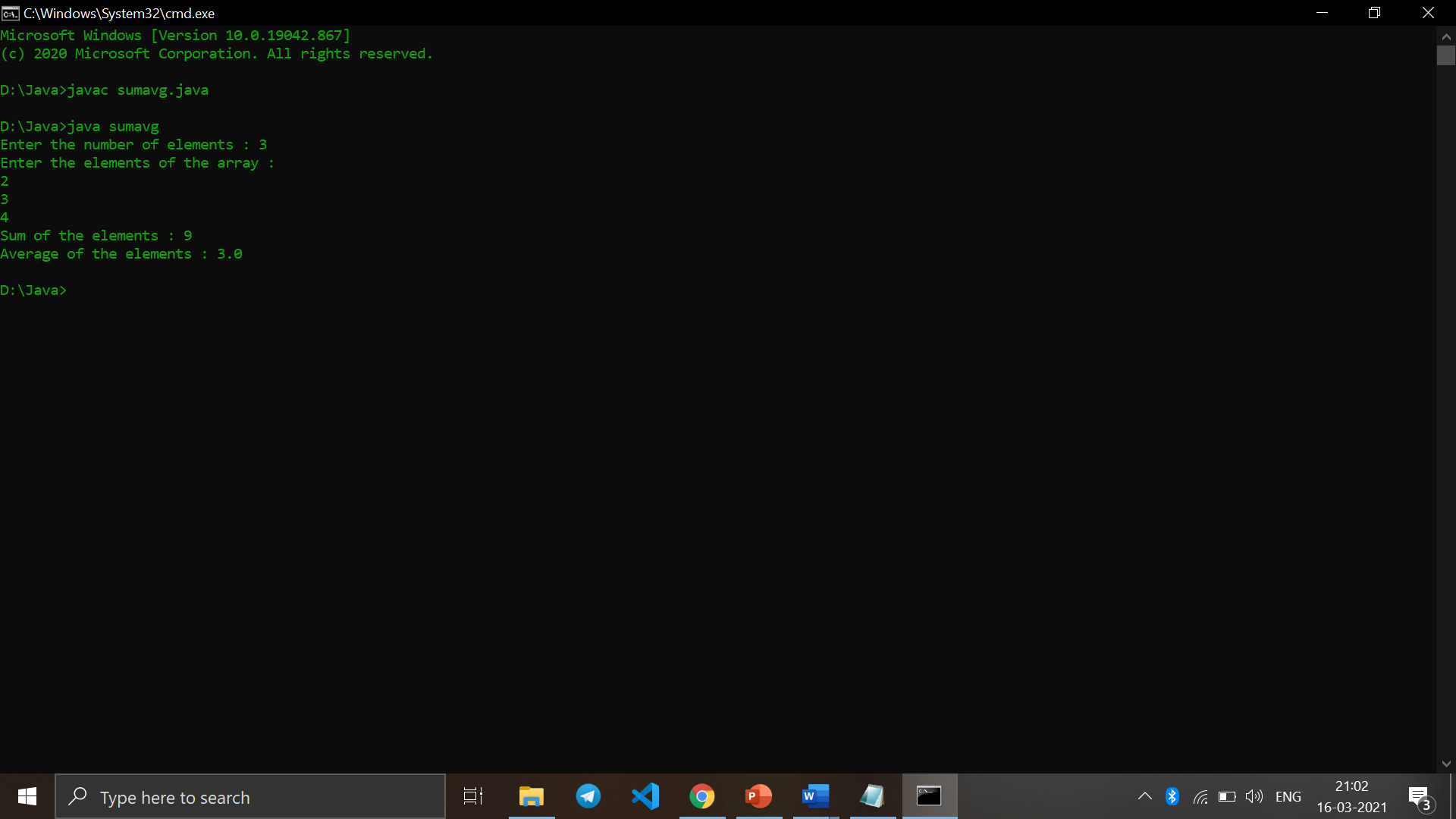
System.out.println("Sum of the elements : "+ s);

System.out.println("Average of the elements : "+ avg);

}

}

Output :



Q2. Write a Java program to implement stack using array.

import java.util.\*;

class Stack

{

int top;

int maxsize = 10;

int[] arr = new int[maxsize];

boolean isEmpty()

{

return (top < 0);

}

Stack()

{

top = -1;

}

boolean push (Scanner in)

{

if(top == maxsize-1)

{

System.out.println("Overflow !!");

return false;

}

else

{

System.out.println("Enter Value");

int val = sc.nextInt();

top++;

arr[top]=val;

System.out.println("Item pushed");

return true;

}

}

boolean pop ()

{

if (top == -1)

{

System.out.println("Underflow !!");

return false;

}

else

{

top --;

System.out.println("Item popped");

return true;

}

}

void display ()

{

System.out.println("Printing stack elements .....");

for(int i = top; i>=0;i--)

{

System.out.println(arr[i]);

}

}

}

public class stac {

public static void main(String[] args) {

int choice=0;

Scanner in = new Scanner(System.in);

Stack s = new Stack();

System.out.println("\*\*\*\*\*\*\*\*\*Stack operations using array\*\*\*\*\*\*\*\*\*\n");

System.out.println("\n------------------------------------------------\n");

while(choice != 4)

{

System.out.println("\nChose one from the below options...\n");

System.out.println("\n1.Push\n2.Pop\n3.Show\n4.Exit");

System.out.println("\n Enter your choice \n");

choice = in.nextInt();

switch(choice)

{

case 1:

{

s.push(in);

break;

}

case 2:

{

s.pop();

break;

}

case 3:

{

s.display();

break;

}

case 4:

{

System.out.println("Exiting....");

System.exit(0);

break;

}

default:

{

System.out.println("Please Enter valid choice ");

}

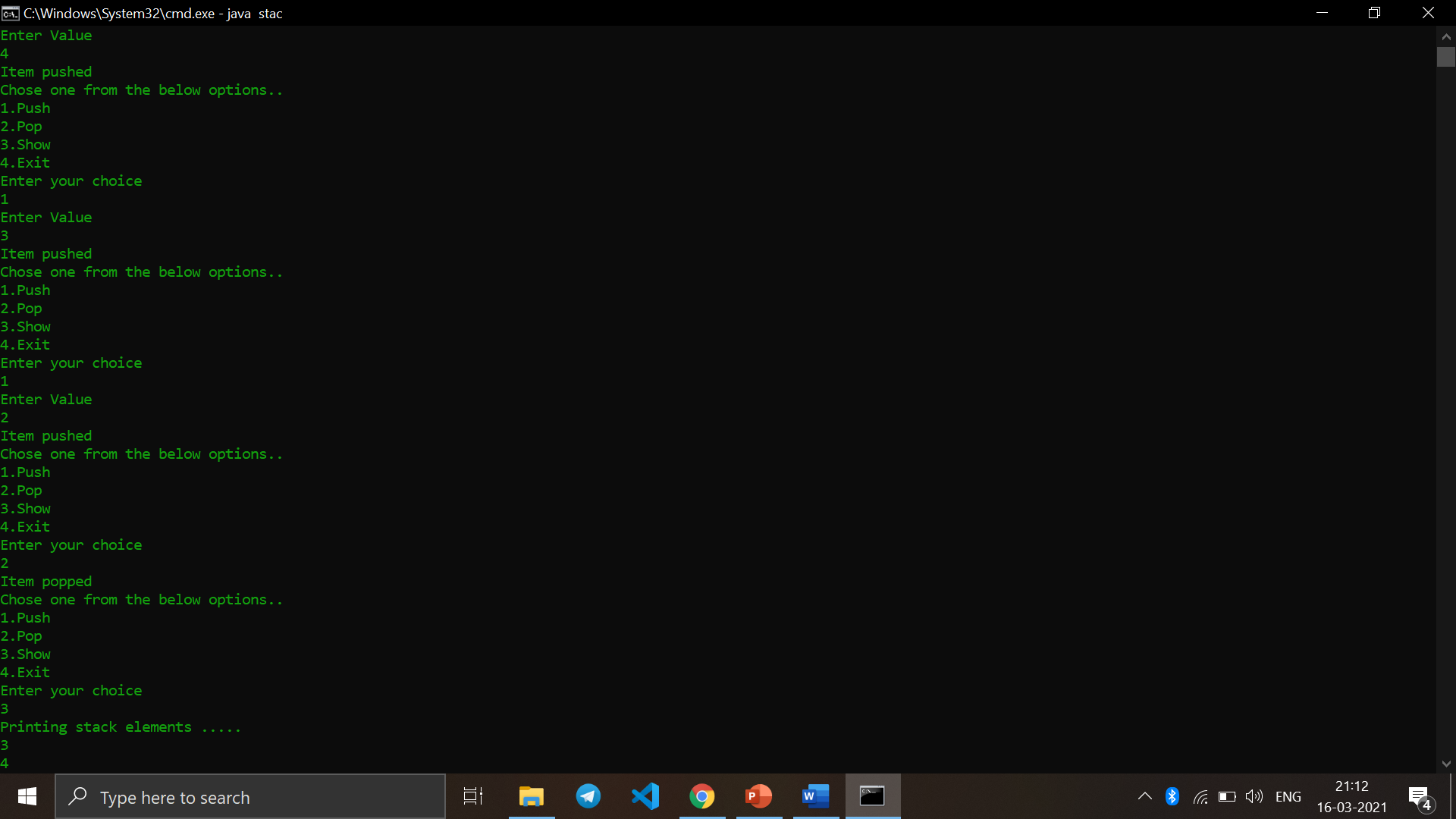
};

}

}

}

Output :



Q3. Write a Java program to implement Queue using array.

import java.io.\*;

class QueueArr

{

static int i,front,rear,item,max=5,ch;

static int a[]=new int[5];

QueueArr()

{

front=-1;

rear=-1;

}

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

{

while((boolean)true)

{

try

{

System.out.println("Select Option 1.insert 2.delete 3.display 4.Exit");

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

ch=Integer.parseInt(br.readLine());

}

catch(Exception e)

{ }

if(ch==4)

break;

else

{

switch(ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

}

}

}

}

static void insert()

{

if(rear>=max)

{

System.out.println("Queue is Full");

}

else

{

try

{

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

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

item=Integer.parseInt(br.readLine());

}

catch(Exception e)

{}

rear=rear+1;

a[rear]=item;

}

}

static void delete()

{

if(front==-1)

{

System.out.println("Queue is Empty");

}

else

{

front=front+1;

item=a[front];

System.out.println("Deleted Item: "+item);

}

}

static void display()

{

System.out.println("Elements in the Queue are:");

for(int i=front+1; i<=rear; i++)

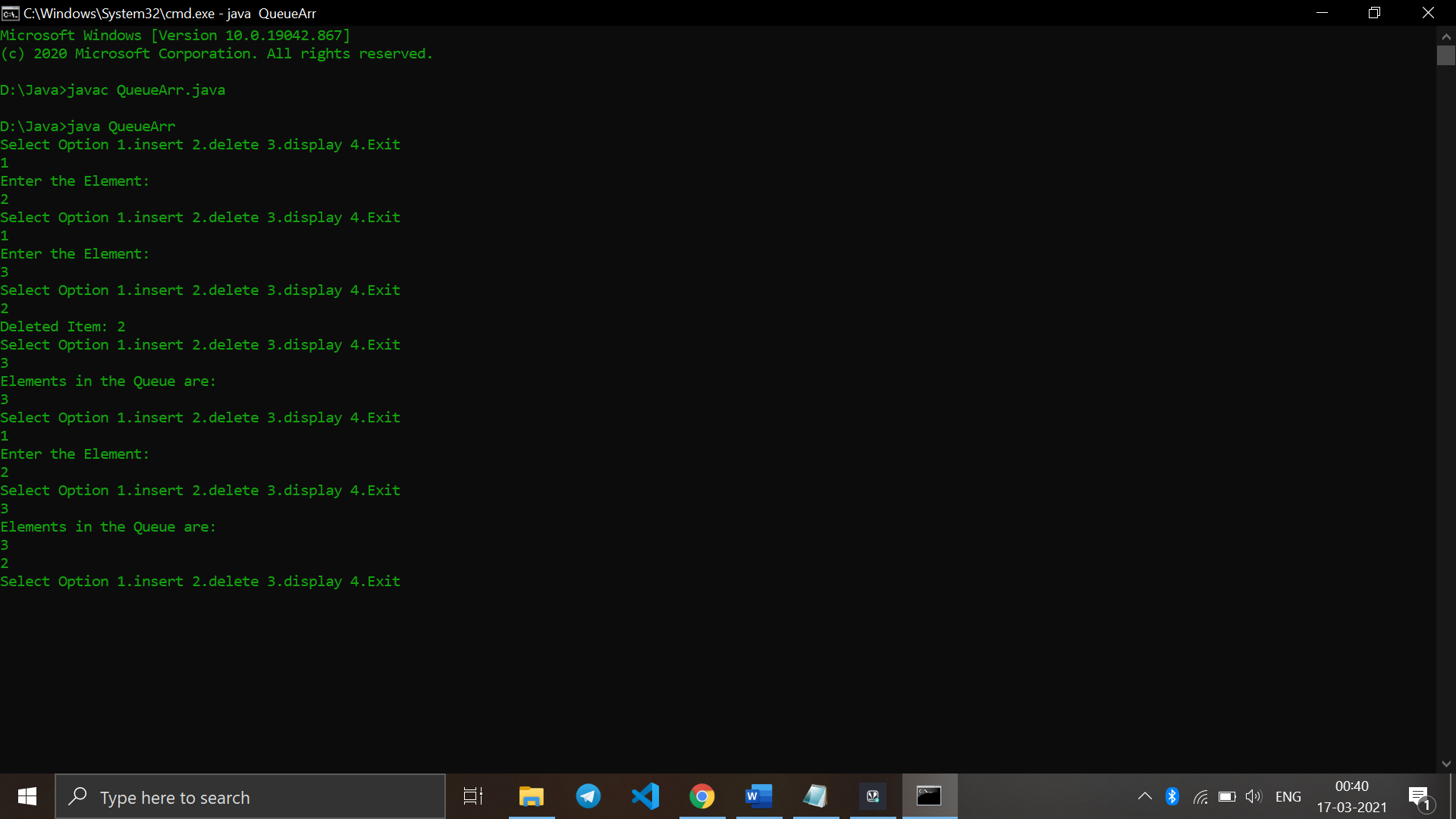
{

System.out.println(a[i]);

}

}

} Output :



Q4.Write a Java program to calculate Sum of two 2-dimensional arrays.

import java.util.Scanner;

public class sum

{

public static void main(String args[])

{

int m, n, c, d;

Scanner in = new Scanner(System.in);

System.out.println("Input number of rows of matrix");

m = in.nextInt();

System.out.println("Input number of columns of matrix");

n = in.nextInt();

int array1[][] = new int[m][n];

int array2[][] = new int[m][n];

int sum[][] = new int[m][n];

System.out.println("Input elements of first matrix");

for ( c = 0 ; c < m ; c++ )

for ( d = 0 ; d < n ; d++ )

array1[c][d] = in.nextInt();

System.out.println("Input the elements of second matrix");

for ( c = 0 ; c < m ; c++ )

for ( d = 0 ; d < n ; d++ )

array2[c][d] = in.nextInt();

for ( c = 0 ; c < m ; c++ )

for ( d = 0 ; d < n ; d++ )

sum[c][d] = array1[c][d] + array2[c][d];

System.out.println("Sum of the matrices:-");

for ( c = 0 ; c < m ; c++ )

{

for ( d = 0 ; d < n ; d++ )

System.out.print(sum[c][d]+"\t");

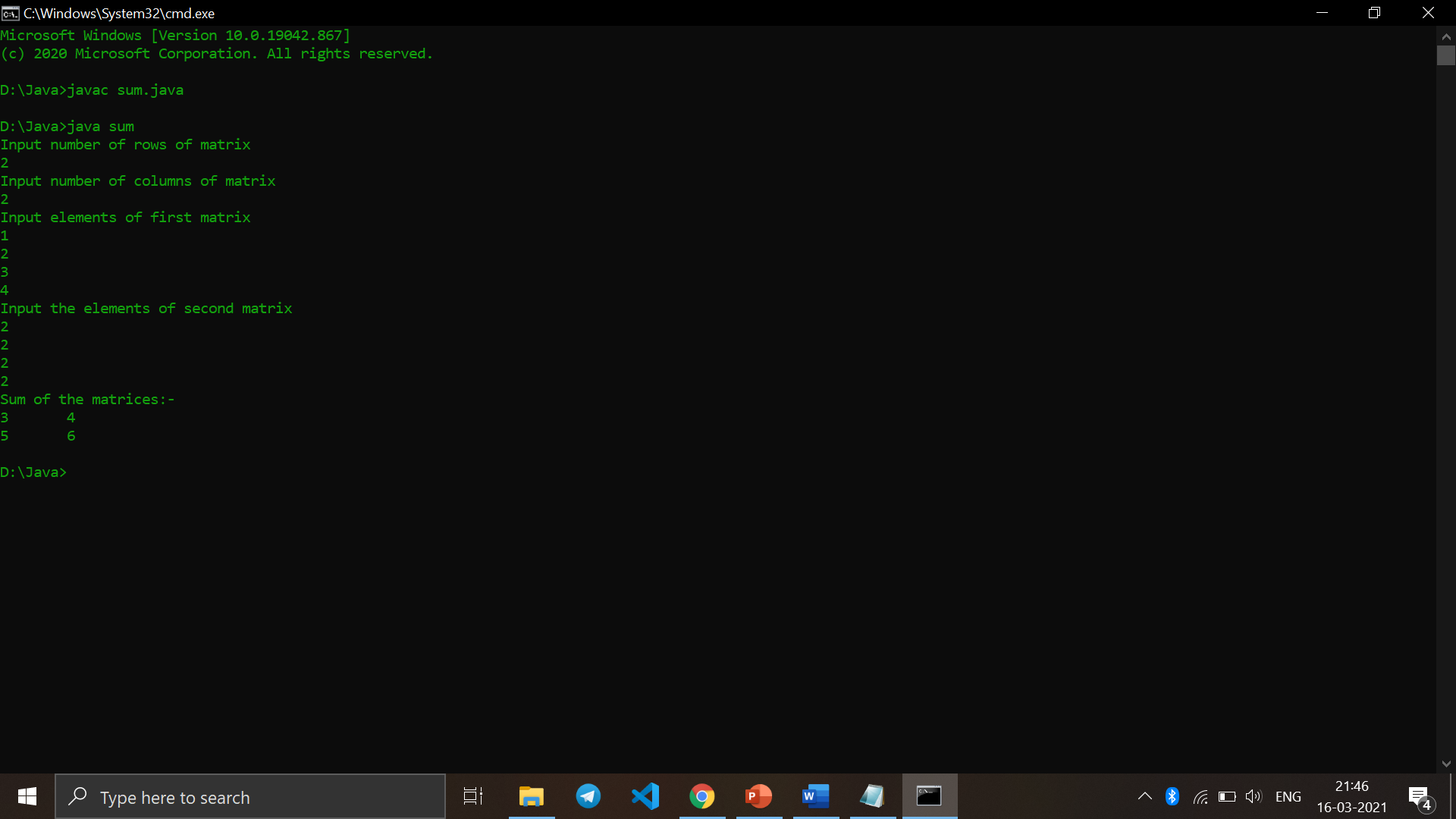
System.out.println();

}

}

}

Output :



Q5. Write a Java program to find the range of a 1D array.

import java.io.\*;

class range {

static float getMin(float arr[], int n)

{

float res = arr[0];

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

res = Math.min(res, arr[i]);

return res;

}

static float getMax(float arr[], int n)

{

float res = arr[0];

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

res = Math.max(res, arr[i]);

return res;

}

static void findRangeAndCoefficient(float arr[], int n)

{

float max = getMax(arr, n);

float min = getMin(arr, n);

float range = max - min;

System.out.println("Range : " + range );

}

public static void main (String[] args) {

float arr[] = { 5, 10, 15, 20, 56, 54 };

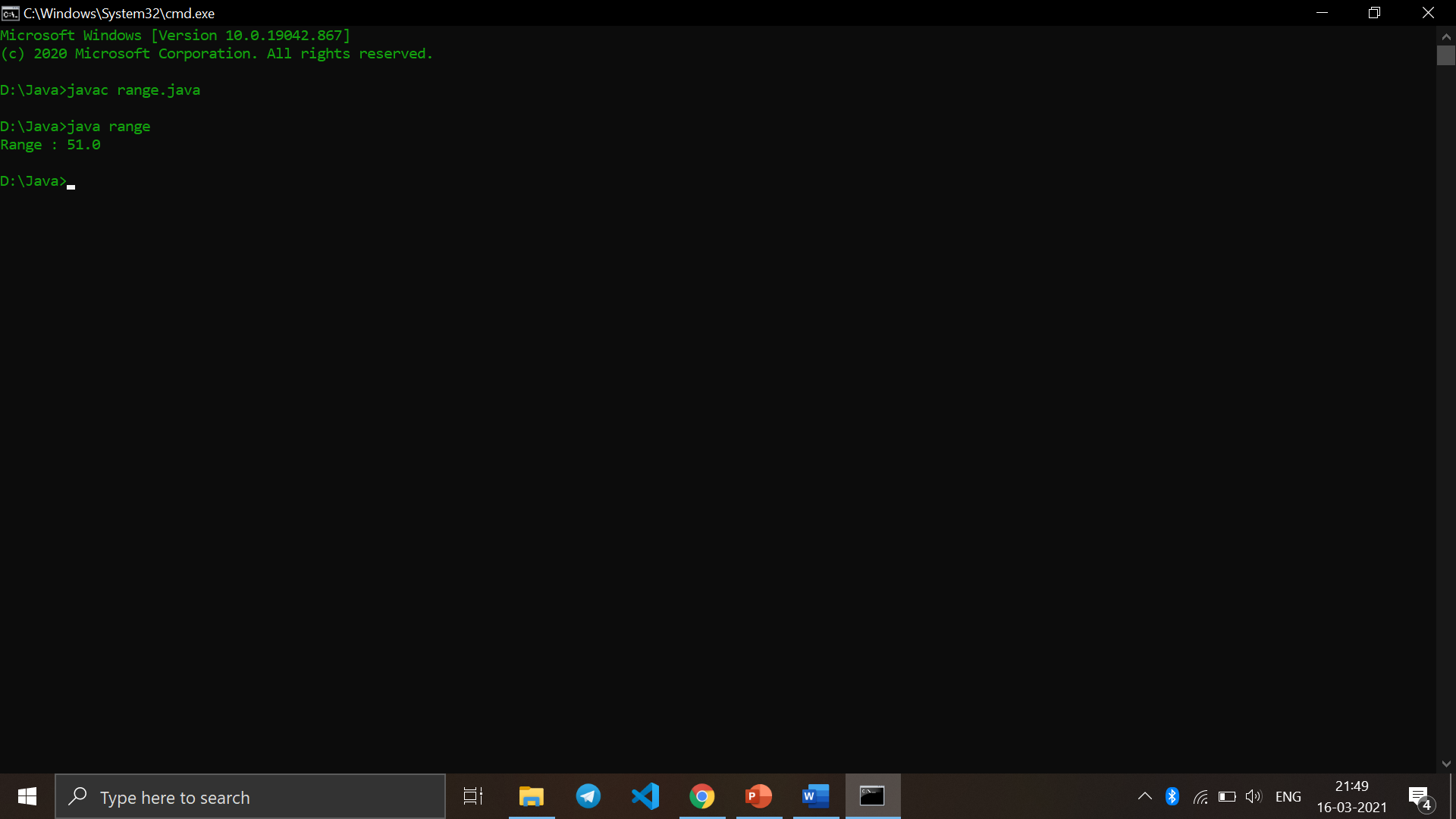
int n = arr.length;

findRangeAndCoefficient(arr, n);

}

}

Output :



Q6. Write a Java program to search an element in an array.

import java.util.Scanner;

class search

{

public static void main(String args[])

{

int c, n, search, array[];

Scanner in = new Scanner(System.in);

System.out.println("Enter number of elements");

n = in.nextInt();

array = new int[n];

System.out.println("Enter those " + n + " elements");

for (c = 0; c < n; c++)

array[c] = in.nextInt();

System.out.println("Enter value to find");

search = in.nextInt();

for (c = 0; c < n; c++)

{

if (array[c] == search) /\* Searching element is present \*/

{

System.out.println(search + " is present at location " + (c + 1) + ".");

break;

}

}

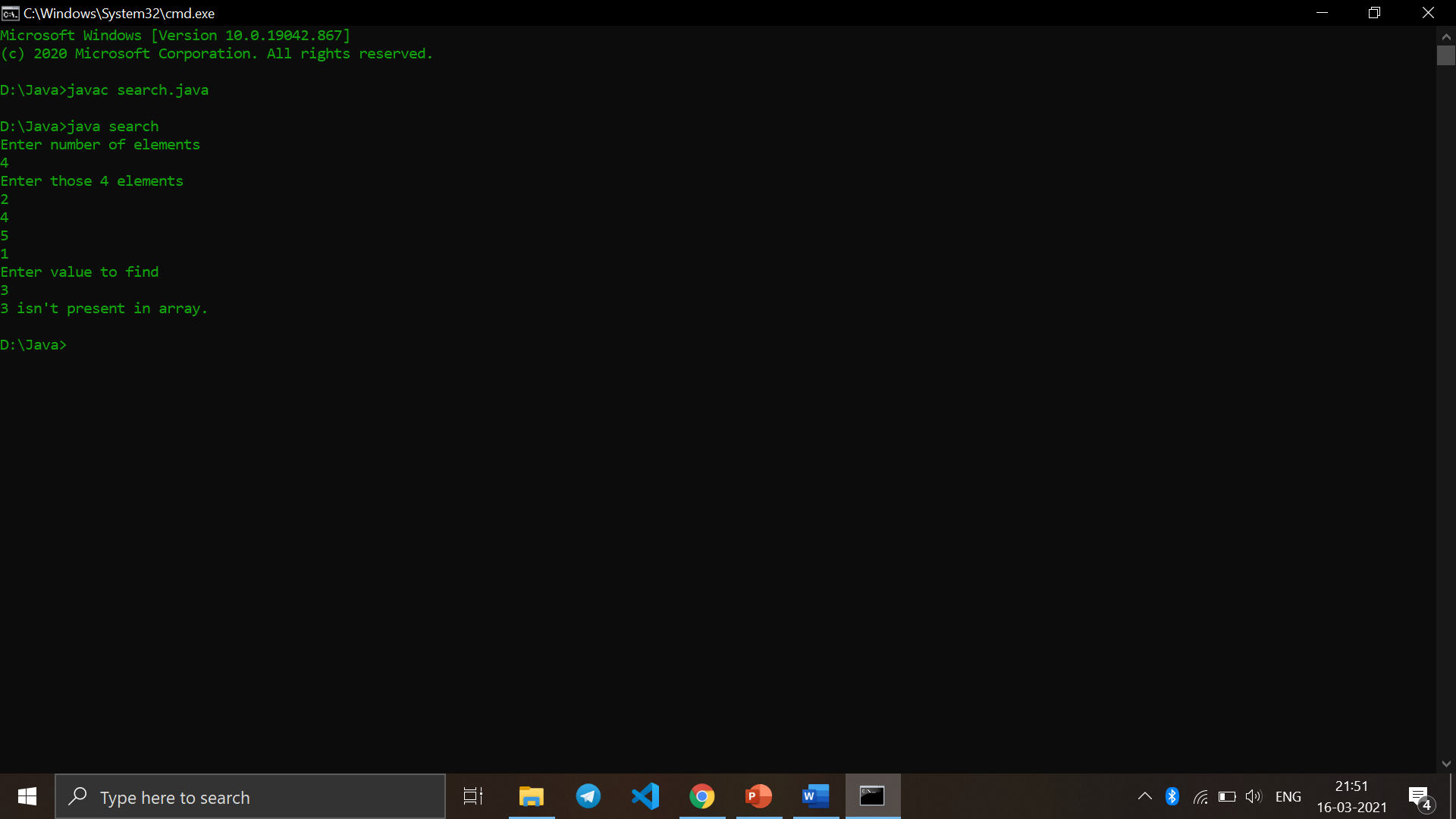
if (c == n) /\* Element to search isn't present \*/

System.out.println(search + " isn't present in array.");

}

}

Output :



Q7. Write a Java program to find the sum of even numbers in an integer array.

import java.util.Scanner;

public class even {

private static Scanner sc;

public static void main(String[] args)

{

int Size, i, EvenSum = 0;

sc = new Scanner(System.in);

System.out.print(" Please Enter Number of elements in an array : ");

Size = sc.nextInt();

int [] a = new int[Size];

System.out.print(" Please Enter " + Size + " elements of an Array : ");

for (i = 0; i < Size; i++)

{

a[i] = sc.nextInt();

}

for(i = 0; i < Size; i++)

{

if(a[i] % 2 == 0)

{

EvenSum = EvenSum + a[i];

}

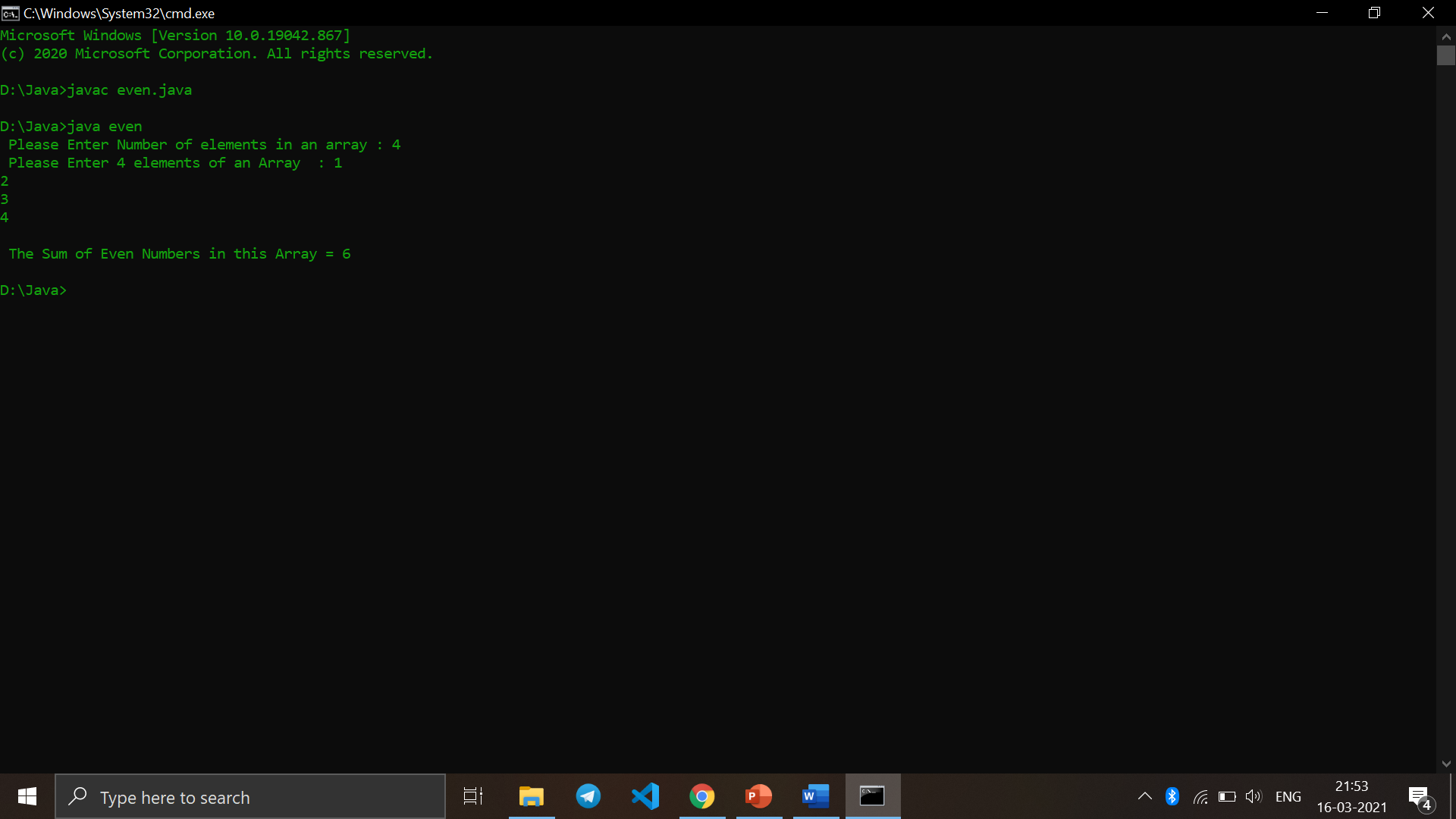
}

System.out.println("\n The Sum of Even Numbers in this Array = " + EvenSum);

}

}

Output :



Q8. Write a Java program to find the sum of diagonal elements in a 2D array.

import java.io.\*;

public class sumd {

static void printDiagonalSums(int [][]mat,

int n)

{

int principal = 0, secondary = 0;

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

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

// Condition for principal

// diagonal

if (i == j)

principal += mat[i][j];

// Condition for secondary

// diagonal

if ((i + j) == (n - 1))

secondary += mat[i][j];

}

}

System.out.println("Principal Diagonal : "

+ principal);

System.out.println("Secondary Diagonal : "

+ secondary);

}

// Driver code

static public void main (String[] args)

{

int [][]a = { { 1, 2, 3, 4 },

{ 5, 6, 7, 8 },

{ 1, 2, 3, 4 },

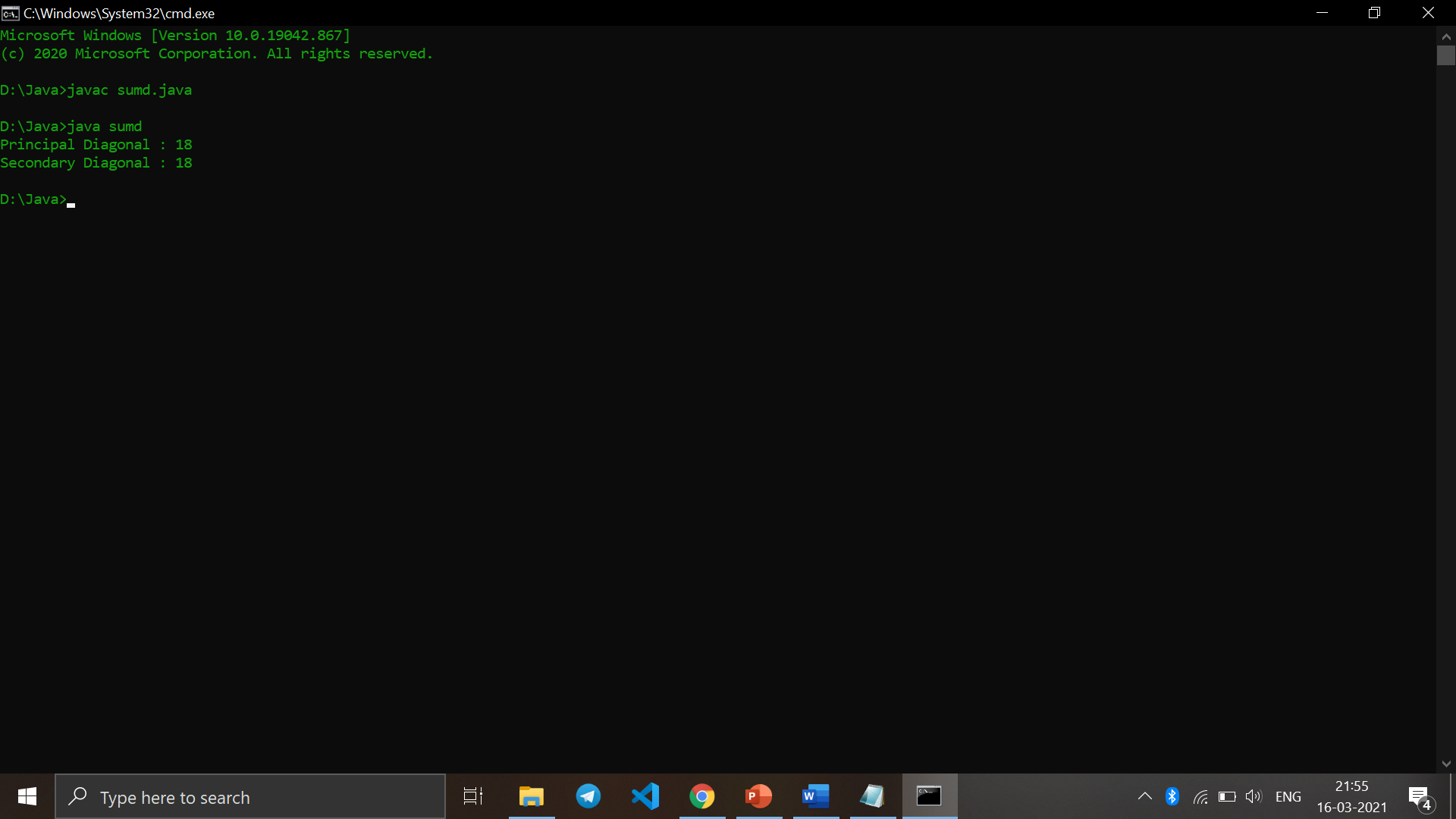
{ 5, 6, 7, 8 } };

printDiagonalSums(a, 4);

}

}

Output :



Q9. Reverse the elements in an array of integers without using a second array.

import java.util.Arrays;

public class rev {

public int[] reverse(int [] array){

if(array == null || array.length <= 1){

System.out.println("Invalid array.");

}

for (int i = 0; i < array.length / 2; i++) {

int temp = array[i];

array[i] = array[array.length - 1 - i];

array[array.length - 1 - i] = temp;

}

return array;

}

public static void main(String[] args){

a3q1 arrayReverse = new a3q1();

int[] input = {1, 2, 3, 4, 5, 6, 7, 8};

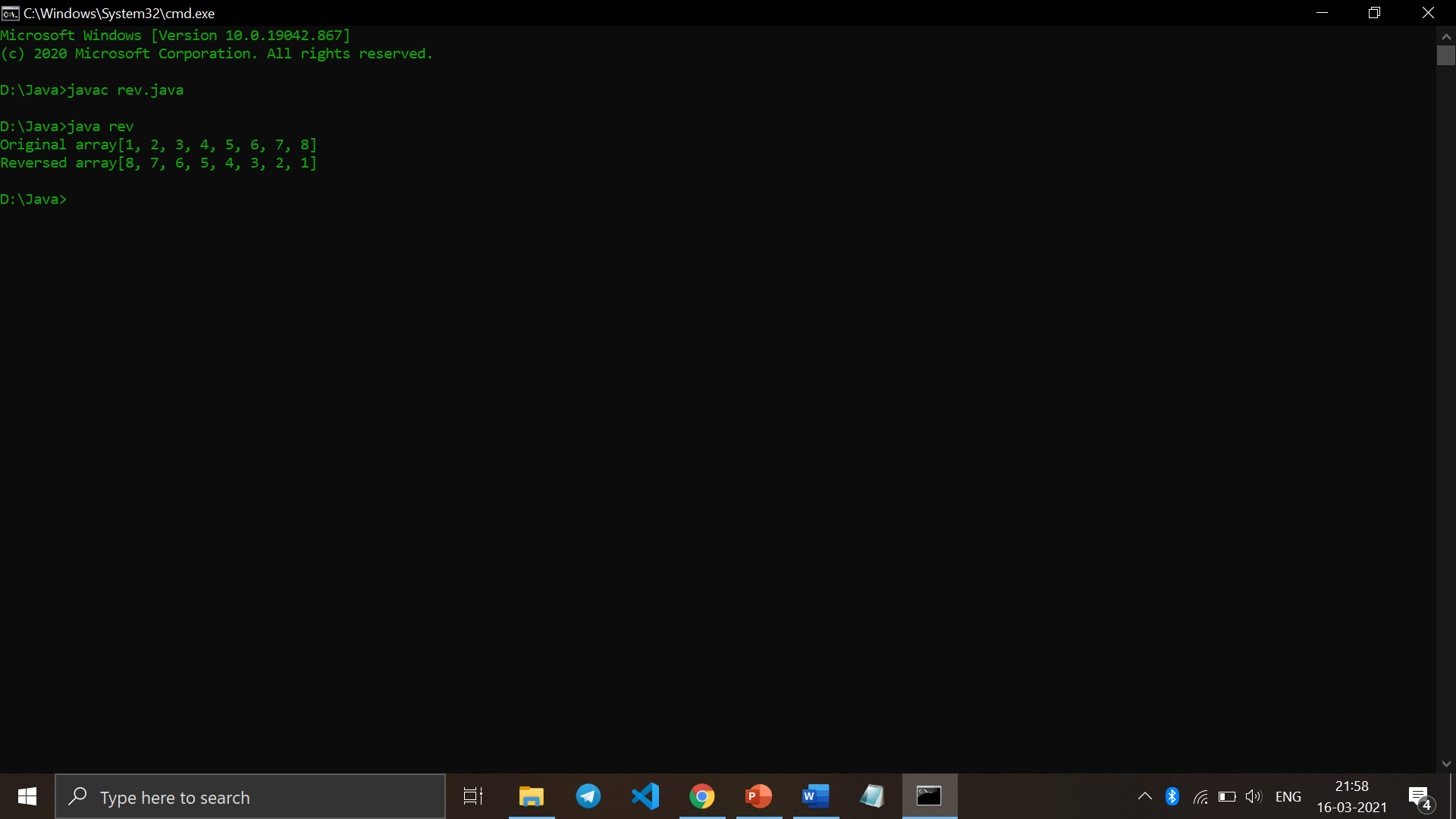
System.out.println("Original array" + Arrays.toString(input));

System.out.println("Reversed array" + Arrays.toString(arrayReverse.reverse(input)));

}

}

Output :



Q10. Write a Java program to enter n elements in an array and find smallest number among them.

import java.util.Scanner;

public class small{

public static int getSmallest(int[] a, int total){

int temp;

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

{

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

{

if (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

return a[0];

}

public static void main(String args[]){

Scanner sc = new Scanner (System.in);

System.out.print("Enter the no. of elements : ");

int z = sc.nextInt();

int[] array = new int[100];

System.out.print("Enter the elements : ");

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

array[i] = sc.nextInt();

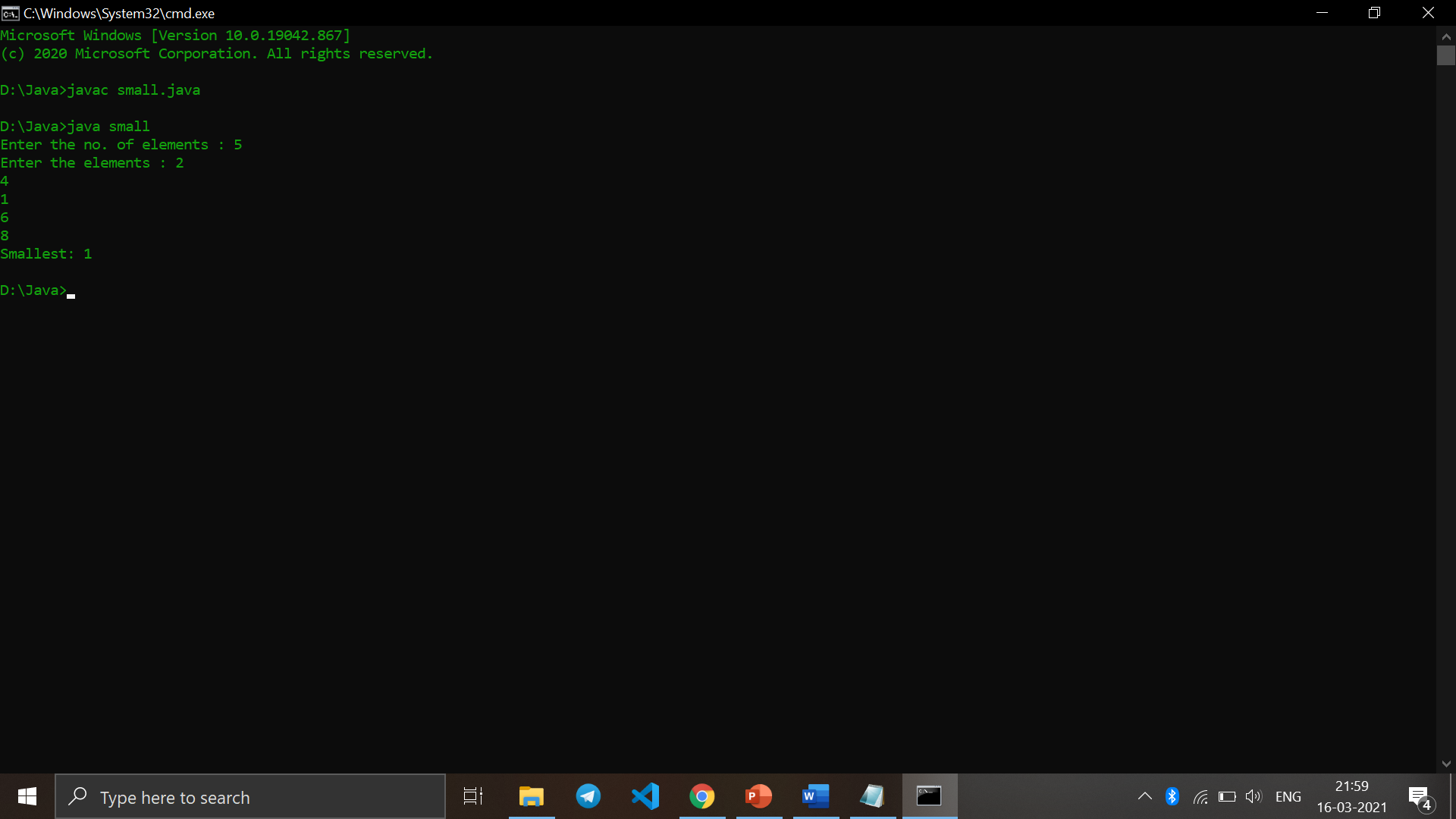
}

System.out.println("Smallest: "+getSmallest(array,z));

}

}

Output :



Q11. Write Java program to find the sum of all odd numbers in a 2D array.

import java.util.Scanner;

public class odd {

public static void main(String[] args)

{

int OddSum = 0;

Scanner sc = new Scanner(System.in);

System.out.print(" Please Enter Number of rows in an array : ");

int row = sc.nextInt();

System.out.print(" Please Enter Number of columns in an array : ");

int col = sc.nextInt();

int [][] a = new int[row][col];

System.out.print(" Please Enter elements of an Array : ");

for (int j=0;j<row;j++){

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

{

a[i][j] = sc.nextInt();

if (a[i][j] %2 != 0){

OddSum = OddSum + a[i][j];

}

}

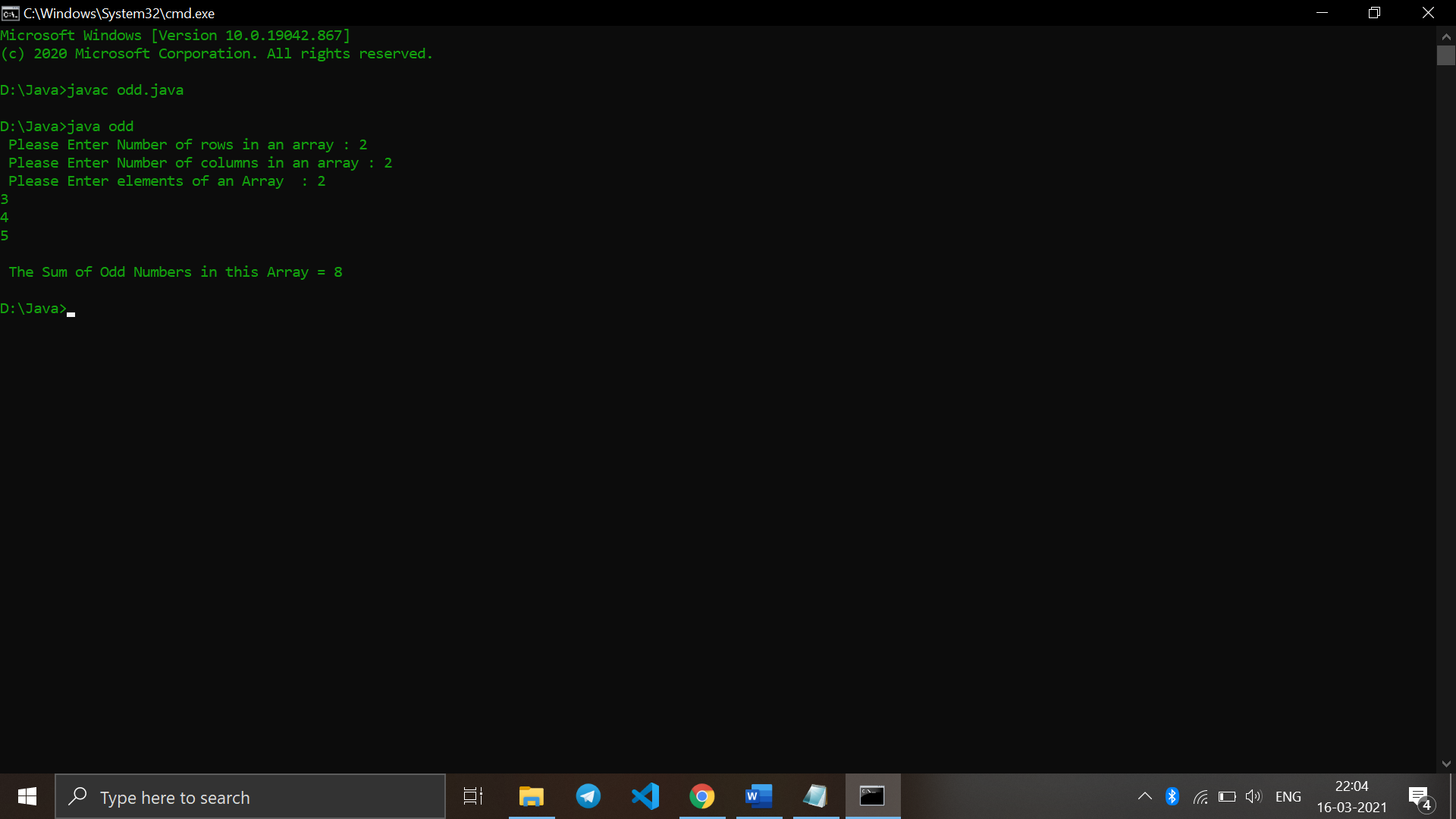
}

System.out.println("\n The Sum of Odd Numbers in this Array = " + OddSum);

}

}

Output :



Q12. Write a Java program to print transpose of matrix.

public class trans{

public static void main(String args[]){

int original[][]={{1,3,4},{2,4,3},{3,4,5}};

int transpose[][]=new int[3][3];

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

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

transpose[i][j]=original[j][i];

}

}

System.out.println("Printing Matrix without transpose:");

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

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

System.out.print(original[i][j]+" ");

}

System.out.println();

}

System.out.println("Printing Matrix After Transpose:");

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

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

System.out.print(transpose[i][j]+" ");

}

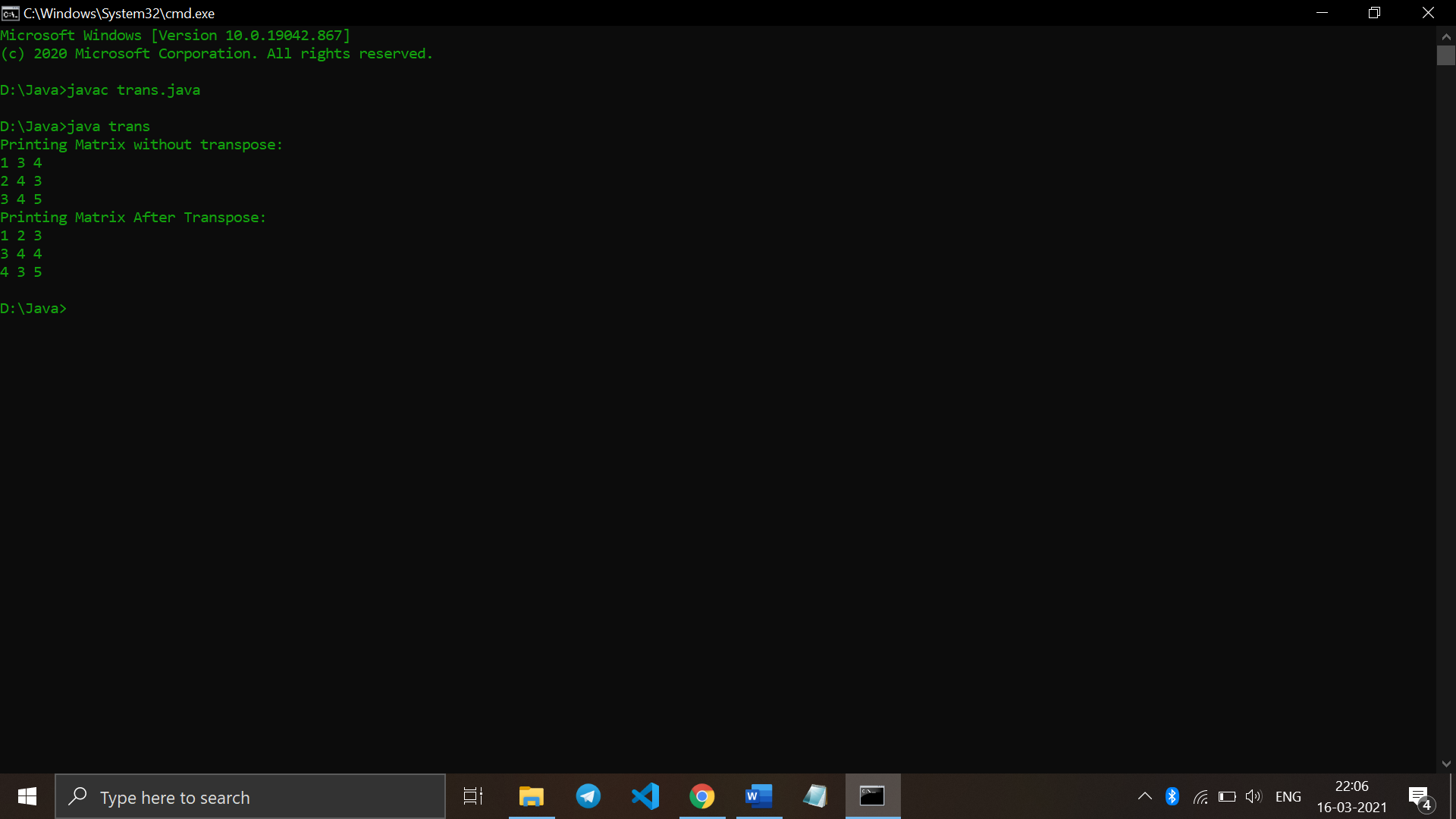
System.out.println();

}

}

}

Output :



Q13. Write a Java program to check whether a given matrix is sparse or not.

import java.util.Scanner;

public class sparse

{

public static void main(String args[])

{

int i, j, zero = 0, count = 0;

int array[][] = new int[10][10];

System.out.println("Enter total rows and columns: ");

Scanner sc = new Scanner(System.in);

int row = sc.nextInt();

int column = sc.nextInt();

System.out.println("Enter matrix:");

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

array[i][j] = sc.nextInt();

System.out.print(" ");

}

}

for(i = 0; i < row; i++)

{

for(j = 0; j < column; j++)

{

if(array[i][j] == 0)

{

zero++;

}

else

{

count++;

}

}

}

if(zero>count)

{

System.out.println("the matrix is sparse matrix");

}

else

{

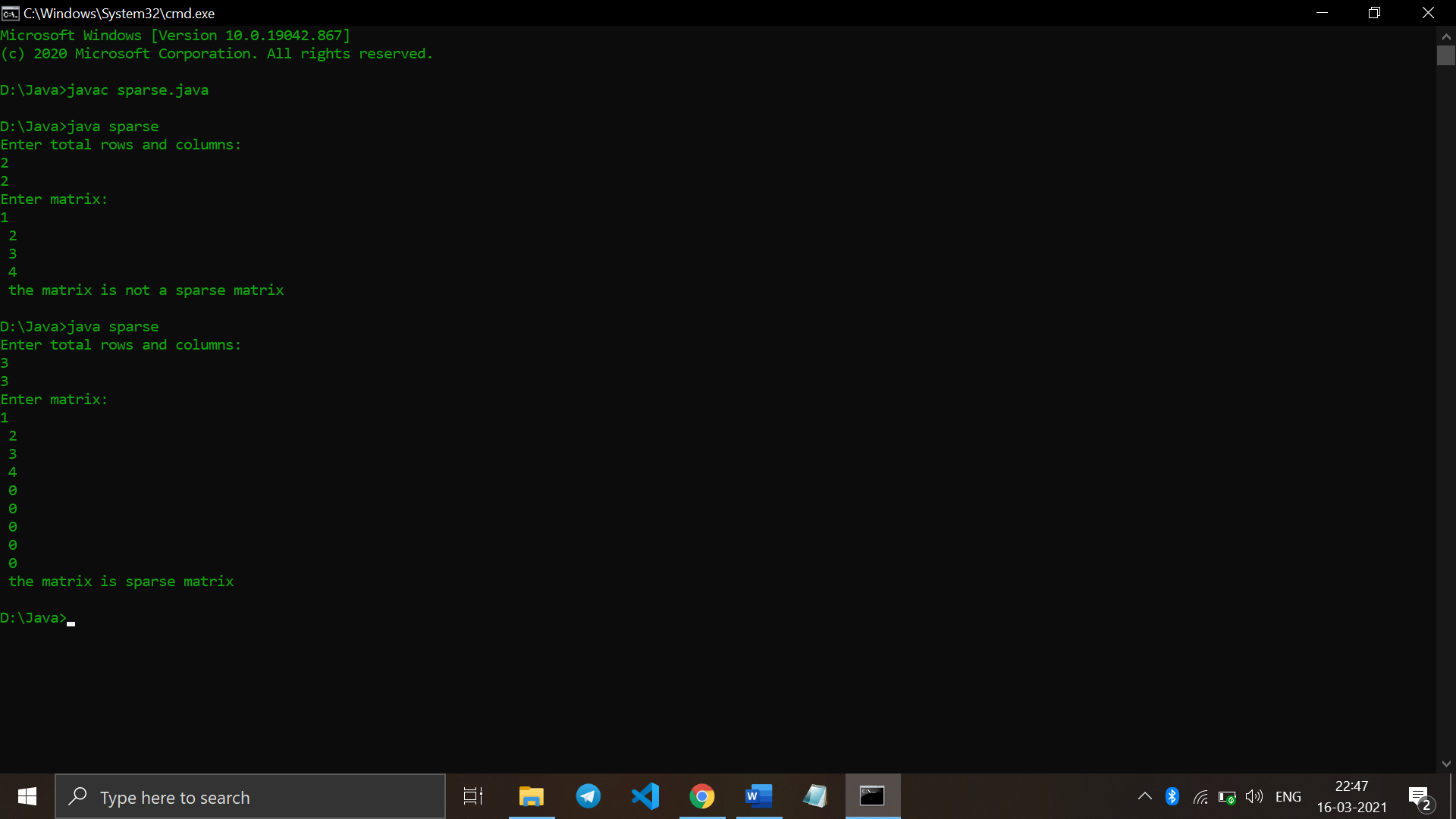
System.out.println("the matrix is not a sparse matrix");

}

}

}

Output :



Q14. Write a Java program to count the prime numbers in an array.

import java.util.Arrays;

import java.util.Vector;

import java.util.\*;

class count

{

static int primeCount(int arr[], int n)

{

int max\_val = Arrays.stream(arr).max().getAsInt();

Boolean[] prime = new Boolean[max\_val + 1];

for (int i = 0; i < max\_val + 1; i++)

{

prime[i] = true;

}

prime[0] = false;

prime[1] = false;

for (int p = 2; p \* p <= max\_val; p++)

{

if (prime[p] == true)

{

for (int i = p \* 2; i <= max\_val; i += p)

{

prime[i] = false;

}

}

}

int count = 0;

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

{

if (prime[arr[i]])

{

count++;

}

}

return count;

}

public static void main(String[] args)

{

Scanner in=new Scanner(System.in);

System.out.println("Enter number of elements");

int n = in.nextInt();

int arr[]= new int[n];

System.out.println("Enter those " + n + " elements");

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

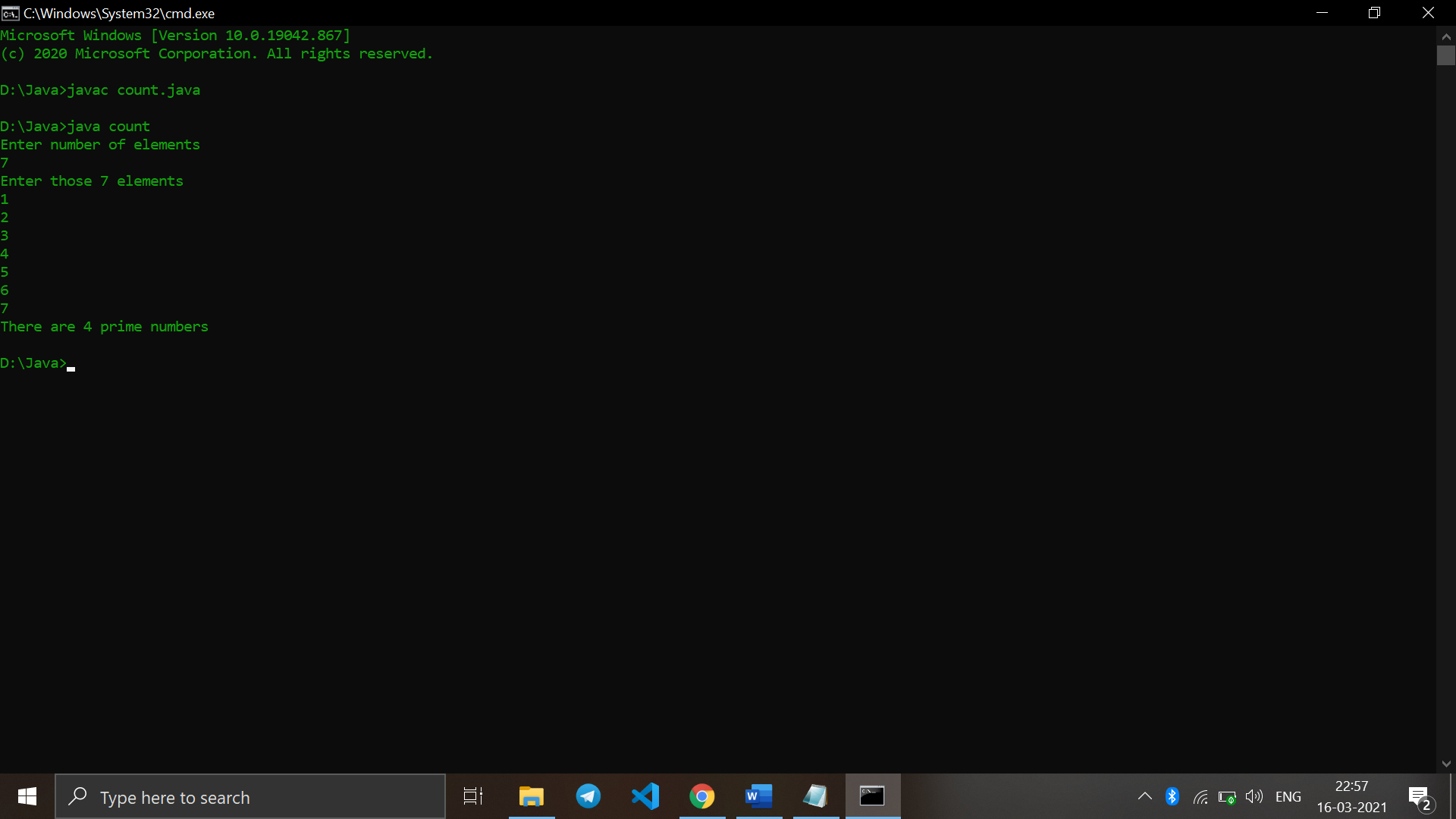
arr[c] = in.nextInt();

System.out.println("There are "+primeCount(arr, n)+" prime numbers");

}

}

Output :



Q15. Write a Java program to find second highest element of an array.

import java.util.\*;

import java.util.Arrays;

class second{

static void print2largest(int arr[],

int arr\_size)

{

int i, first, second;

if (arr\_size < 2)

{

System.out.println(" Invalid Input ");

return;

}

Arrays.sort(arr);

for (i = arr\_size - 2; i >= 0; i--)

{

if (arr[i] != arr[arr\_size - 1])

{

System.out.println("The second largest element is "+arr[i]);

return;

}

}

System.out.println("There is no second largest element");

}

// Driver code

public static void main(String[] args)

{

Scanner in=new Scanner(System.in);

System.out.println("Enter number of elements");

int n = in.nextInt();

int arr[]= new int[n];

System.out.println("Enter those " + n + " elements");

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

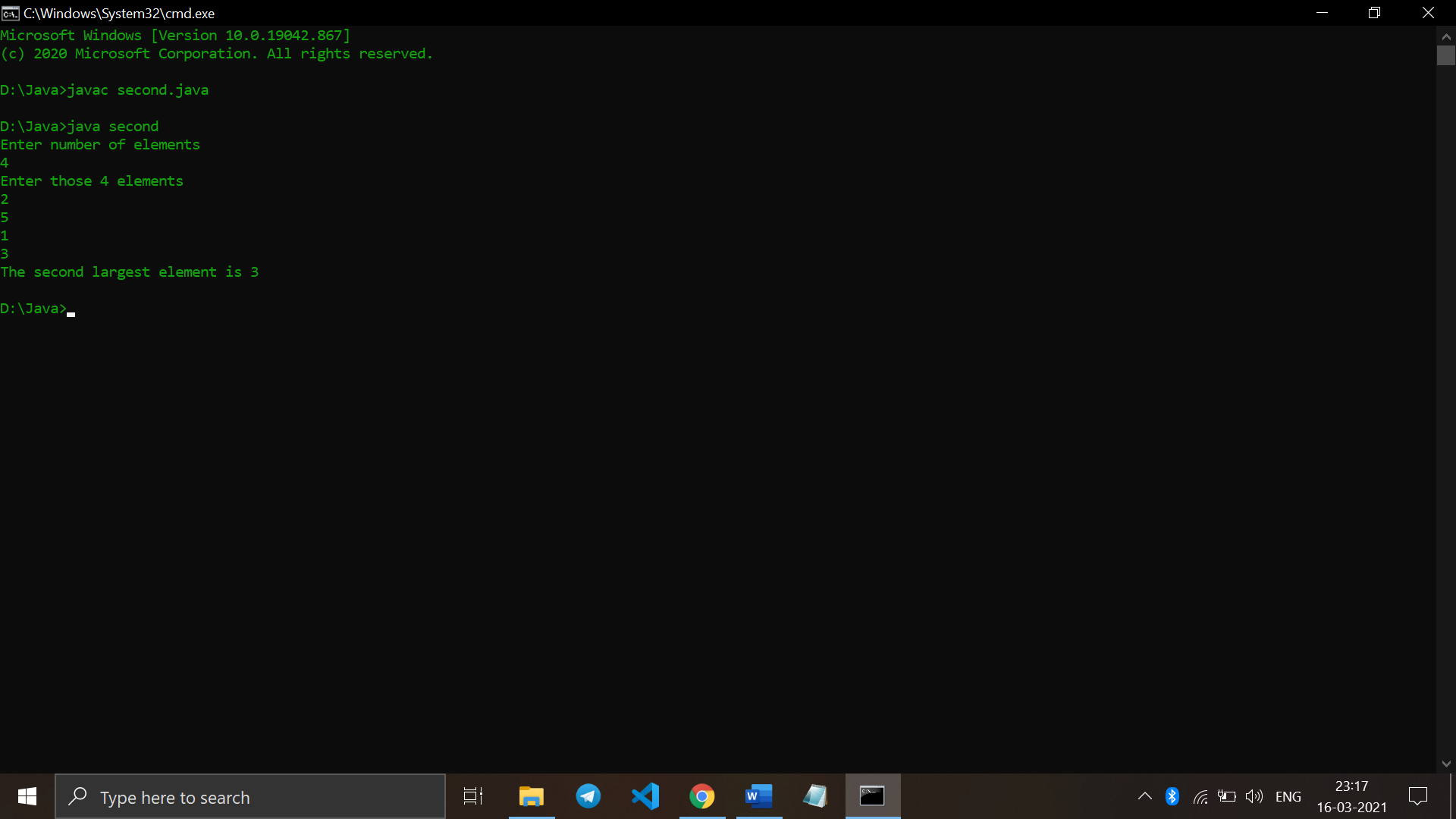
arr[c] = in.nextInt();

print2largest(arr, n);

}

}

Output :



Q16. Write a Java program which counts the non-zero elements in an integer array.

import java.util.\*;

class non{

public static void main(String args[]) {

Scanner sc = new Scanner (System.in);

System.out.print("Enter the no. of elements : ");

int n = sc.nextInt();

int count = 0;

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

int [] arr = new int[100];

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

arr[i] = sc.nextInt();

if (arr[i]!=0){

count++;

}

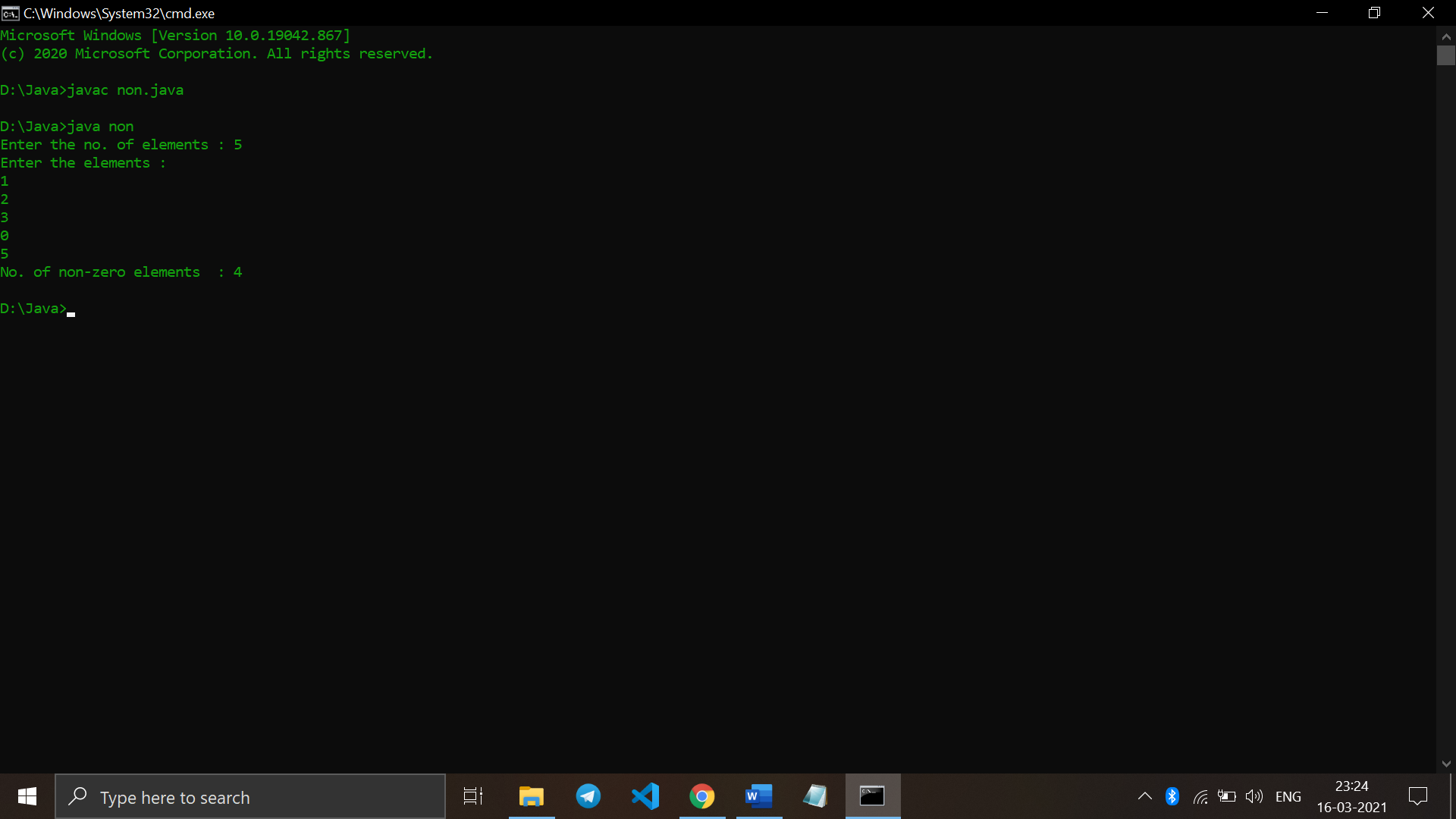
}

System.out.println("No. of non-zero elements : " + count);

}

}

Output :



Q17. Write a Java program to merge two float arrays.

import java.util.Arrays;

public class merge

{

public static void main(String[] args)

{

float[] firstArray = {23f,45f,12f,78f,4f,90f,1f};

float[] secondArray = {77f,11f,45f,88f,32f,56f,3f};

int fal = firstArray.length;

int sal = secondArray.length;

float[] result = new float[fal + sal];

System.arraycopy(firstArray, 0, result, 0, fal);

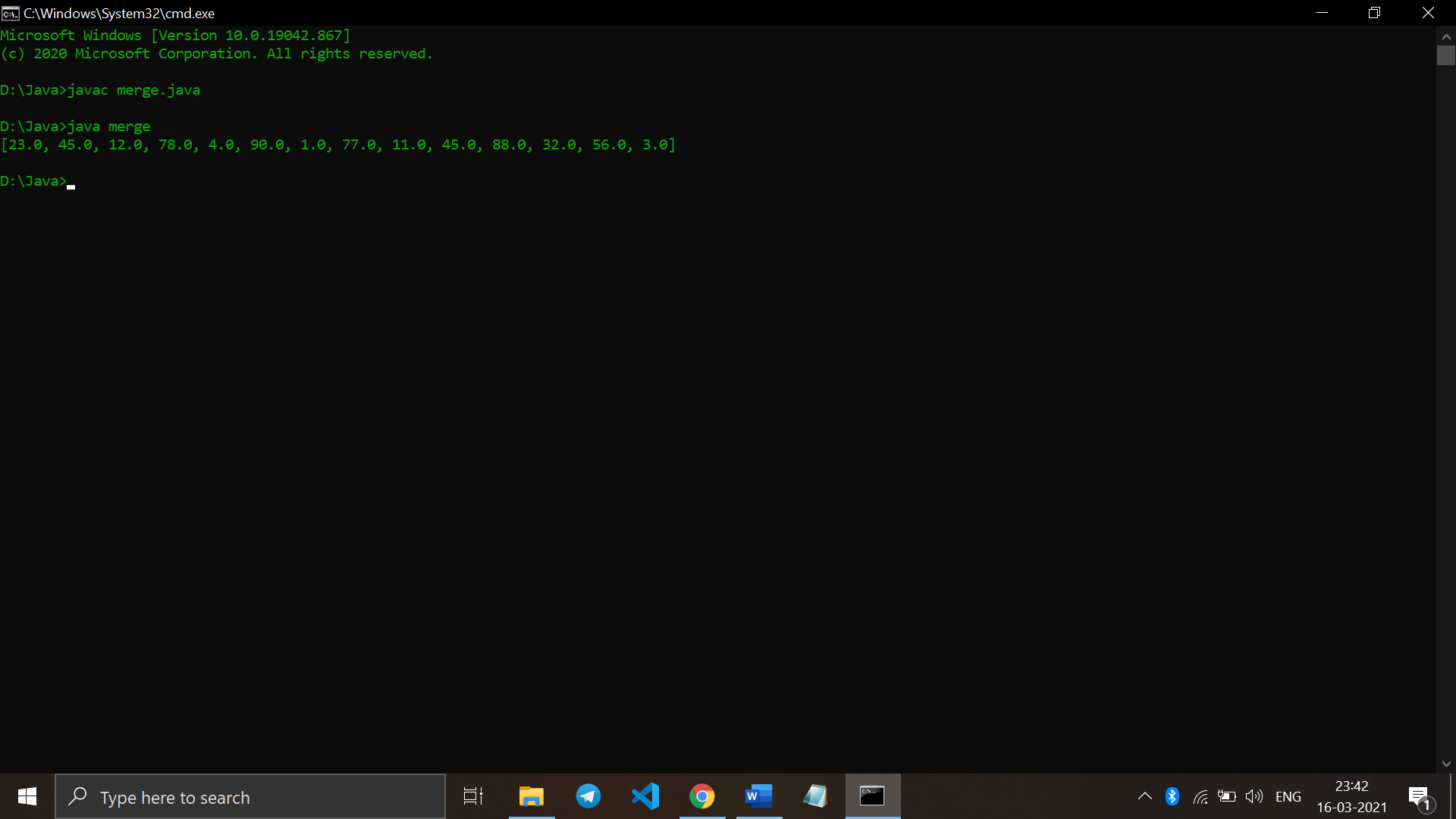
System.arraycopy(secondArray, 0, result, fal, sal);

System.out.println(Arrays.toString(result));

}

}

Output :



Q18. Write a Java program where elements of two integer arrays get added index wise and get stored into a third array.

import java.util.\*;

class add{

public static void main(String args[]) {

Scanner sc = new Scanner (System.in);

System.out.print("Enter the no. of elements of the 1st Array : ");

int n1 = sc.nextInt();

int[] arr1 = new int[100];

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

for (int i = 0; i< n1;i++){

arr1[i] = sc.nextInt();

}

System.out.print("Enter the no. of elements of the 2nd Array : ");

int n2 = sc.nextInt();

int[] arr2 = new int[100];

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

for (int i = 0; i< n2;i++){

arr2[i] = sc.nextInt();

}

int[] result = new int[100];

for (int i = 0; i<n1;i++){

result[i] = arr1[i]+arr2[i];

}

System.out.print("The resultant array : ");

for (int i = 0; i<n1;i++){

System.out.print(result[i]);

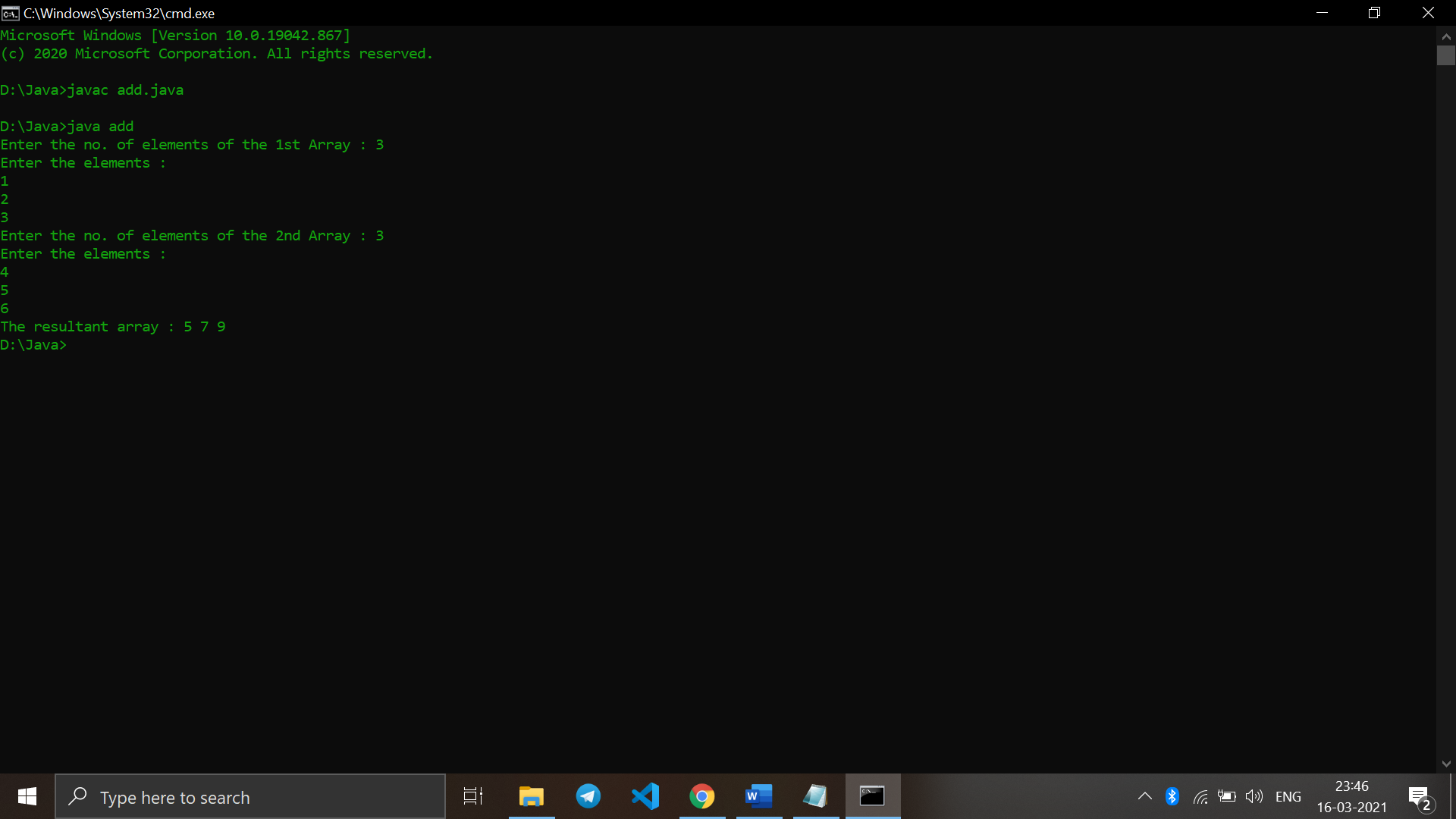
System.out.print(" ");

}

}

}

Output :



Q19. Write a Java program to multiply two matrices.

import java.util.Scanner;

class multi

{

public static void main(String args[])

{

int m, n, p, q, sum = 0, c, d, k;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows and columns of first matrix");

m = in.nextInt();

n = in.nextInt();

int first[][] = new int[m][n];

System.out.println("Enter elements of first matrix");

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

first[c][d] = in.nextInt();

System.out.println("Enter the number of rows and columns of second matrix");

p = in.nextInt();

q = in.nextInt();

if (n != p)

System.out.println("The matrices can't be multiplied with each other.");

else

{

int second[][] = new int[p][q];

int multiply[][] = new int[m][q];

System.out.println("Enter elements of second matrix");

for (c = 0; c < p; c++)

for (d = 0; d < q; d++)

second[c][d] = in.nextInt();

for (c = 0; c < m; c++)

{

for (d = 0; d < q; d++)

{

for (k = 0; k < p; k++)

{

sum = sum + first[c][k]\*second[k][d];

}

multiply[c][d] = sum;

sum = 0;

}

}

System.out.println("Product of the matrices:");

for (c = 0; c < m; c++)

{

for (d = 0; d < q; d++)

System.out.print(multiply[c][d]+"\t");

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

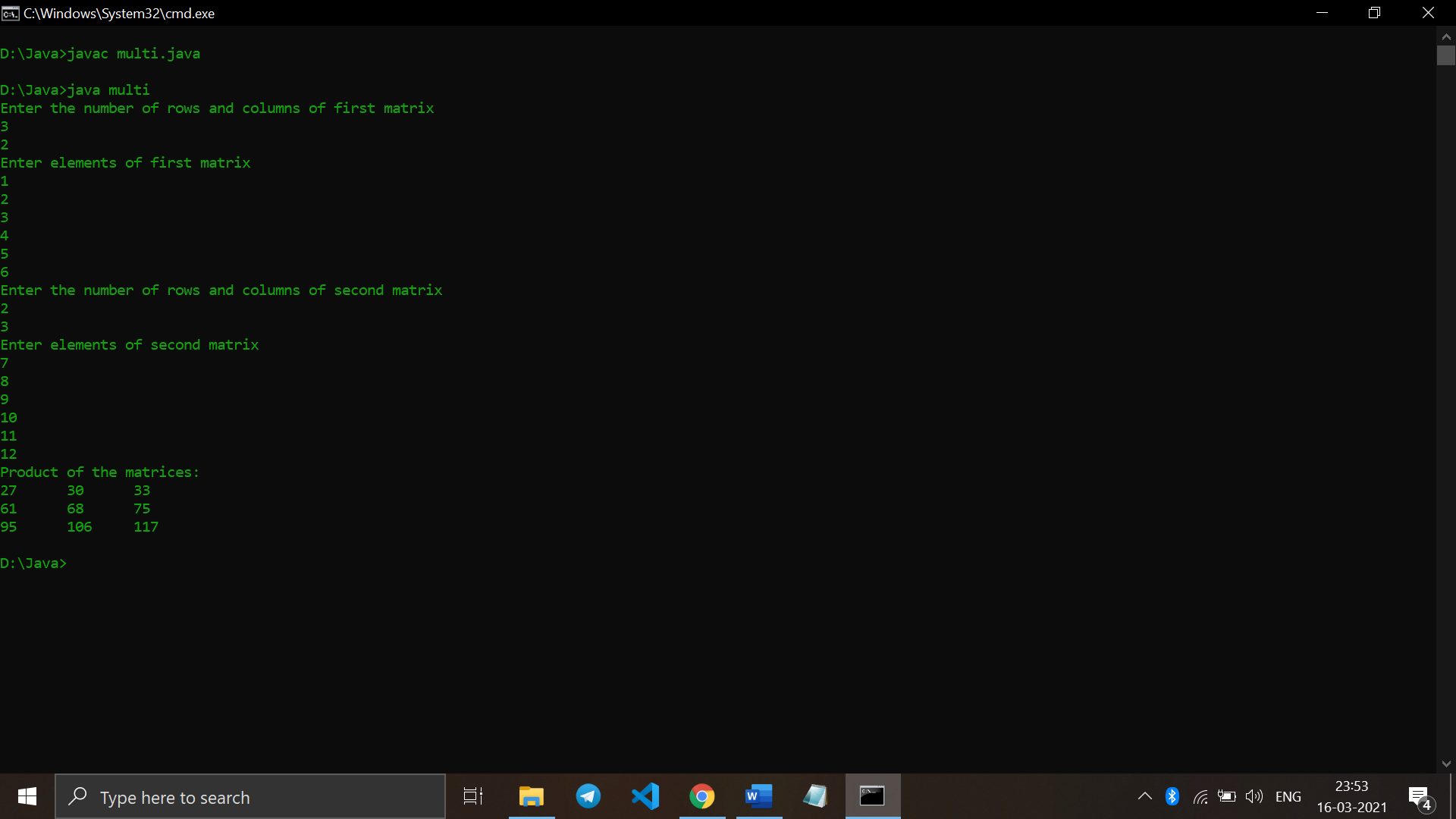
}

}

}

}

Output :



Q20. Write a Java program to subtract two matrices.

import java.util.Scanner;

public class sub

{

public static void main(String args[])

{

int i, j;

System.out.print("Enter the rows and columns of matrices : ");

Scanner scan = new Scanner(System.in);

int r = scan.nextInt();

int c = scan.nextInt();

int mat1[][] = new int[r][c];

int mat2[][] = new int[r][c];

int mat3[][] = new int[r][c];

System.out.print("Enter Matrix 1 Elements : ");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

mat1[i][j] = scan.nextInt();

}

}

System.out.print("Enter Matrix 2 Elements : ");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

mat2[i][j] = scan.nextInt();

}

}

System.out.print("Subtracting Matrices (i.e. Matrix1 - Matrix2)...\n");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

mat3[i][j] = mat1[i][j] - mat2[i][j];

}

}

System.out.print("Result of Matrix1 - Matrix2 is :\n");

for(i=0; i<r; i++)

{

for(j=0; j<c; j++)

{

System.out.print(mat3[i][j]+ " ");

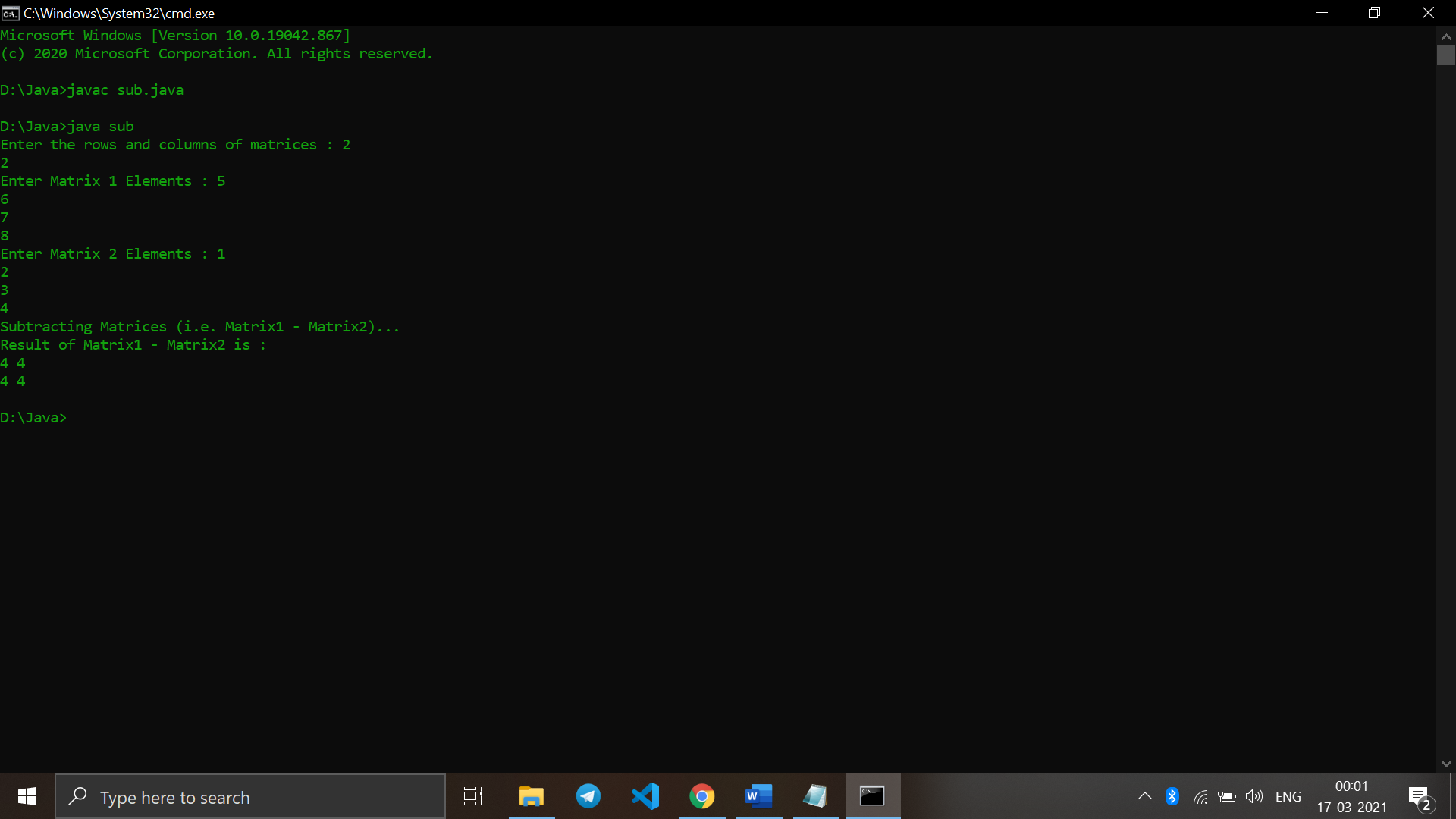
}

System.out.println();

}

}

}Output :



Q21. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence.

import java.util.\*;

import java.util.Arrays;

class duplicate{

public static void findCounts(int arr[], int n)

{

int hash[] = new int[n];

Arrays.fill(hash, 0);

int i = 0;

while (i < n)

{

hash[arr[i] - 1]++;

i++;

}

System.out.println("\nDuplicate elements -> Occurrence");

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

{

if (hash[i]>1){

System.out.println((i + 1) + " -> " +

hash[i]);

}

}

}

public static void main(String []args)

{

Scanner sc = new Scanner (System.in);

System.out.print("Enter the no. of elements : ");

int n = sc.nextInt();

int[] arr = new int[n];

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

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

arr[i] = sc.nextInt();

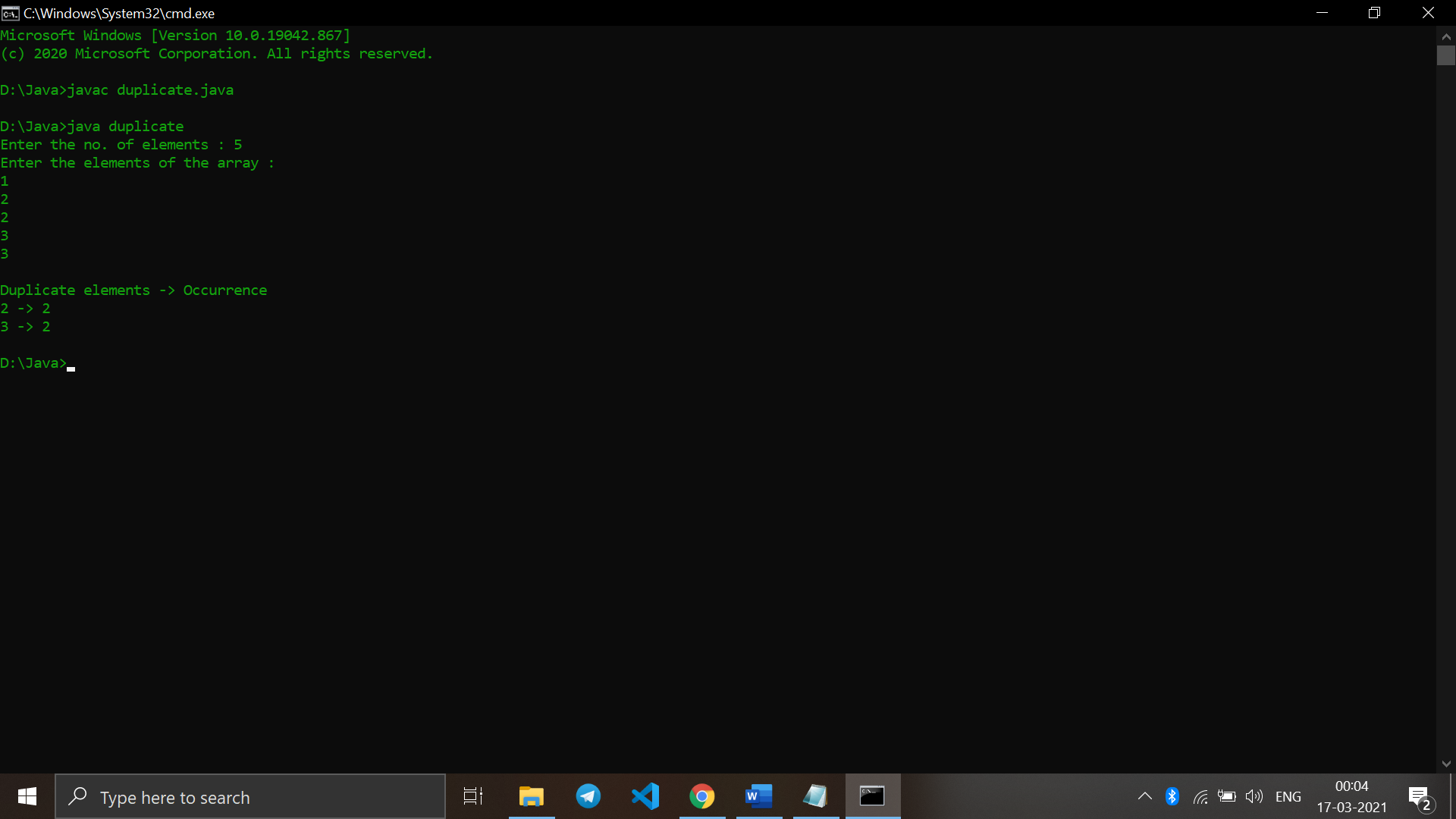
}

findCounts(arr, n);

}

}

Output :



Q22. Write a Java program to print every alternate number of a given array.

import java.util.\*;

import java.io.\*;

class alternate{

static void printAlter(int[] arr, int N)

{

for(int currIndex = 0; currIndex < N; currIndex++)

{

if (currIndex % 2 == 0)

{

System.out.print(arr[currIndex] + " ");

}

}

}

public static void main(String[] args)

{

Scanner sc = new Scanner (System.in);

System.out.print("Enter the no. of elements : ");

int N = sc.nextInt();

int[] arr = new int[N];

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

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

arr[i] = sc.nextInt();

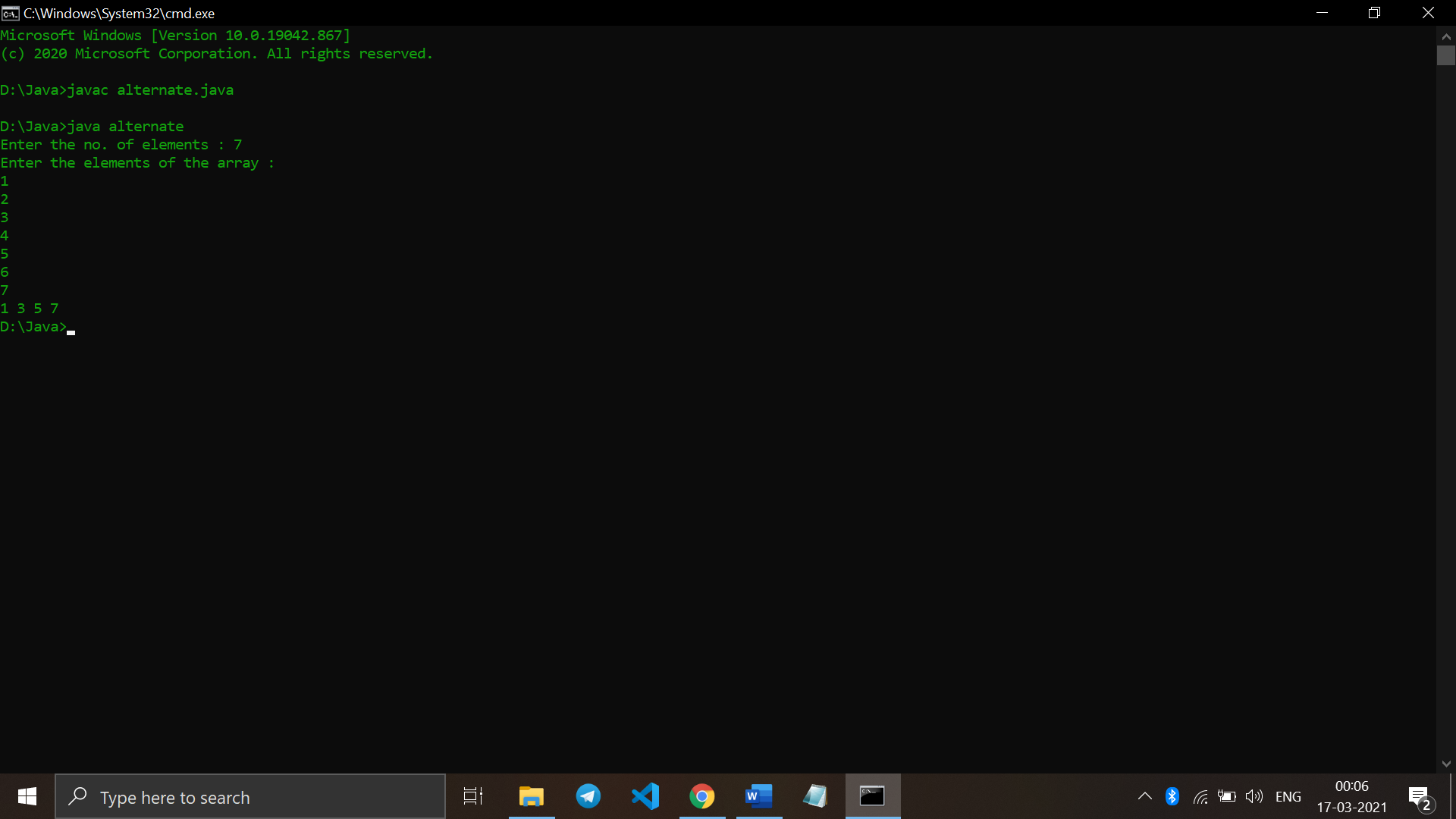
}

printAlter(arr, N);

}

}

Output :



Q23. . Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class mergeasc

{

public static void mergeArrays(int[] arr1, int[] arr2, int n1,

int n2, int[] arr3)

{

int i = 0, j = 0, k = 0;

while (i<n1 && j <n2)

{

if (arr1[i] < arr2[j])

arr3[k++] = arr1[i++];

else

arr3[k++] = arr2[j++];

}

while (i < n1)

arr3[k++] = arr1[i++];

while (j < n2)

arr3[k++] = arr2[j++];

}

public static void main (String[] args)

{

Scanner sc = new Scanner (System.in);

System.out.print("Enter the no. of elements of the first array (A) : ");

int n1 = sc.nextInt();

System.out.print("Enter the no. of elements of the second array (B) : ");

int n2 = sc.nextInt();

int[] arr1 = new int[n1];

int[] arr2 = new int[n2];

System.out.print("Enter the sorted elements of the first array (A) : ");

for (int i=0;i<n1;i++){

arr1[i] = sc.nextInt();

}

System.out.print("Enter the sorted elements of the second array (B) : ");

for (int i=0;i<n2;i++){

arr2[i] = sc.nextInt();

}

int[] arr3 = new int[n1+n2];

mergeArrays(arr1, arr2, n1, n2, arr3);

System.out.print("\nArray after merging (C) in sorted manner : ");

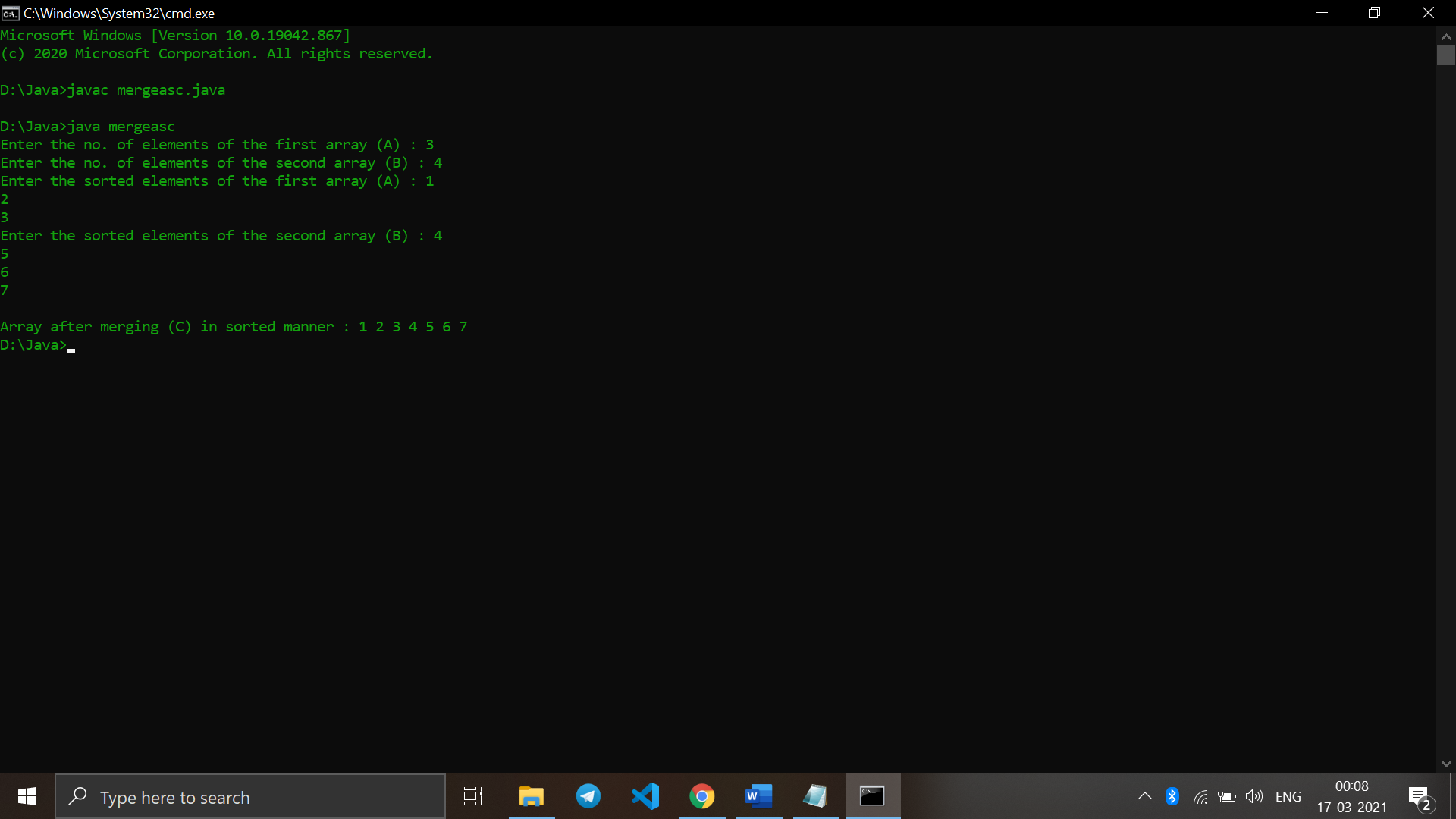
for (int i=0; i < n1+n2; i++)

System.out.print(arr3[i] + " ");

}

}

Output :



Q24. Write a Java program to show 0-arguments constructor.

class zero{

int i;

zero() {

i = 5;

System.out.println("Constructor is called");

}

public static void main(String[] args) {

// calling the constructor without any parameter

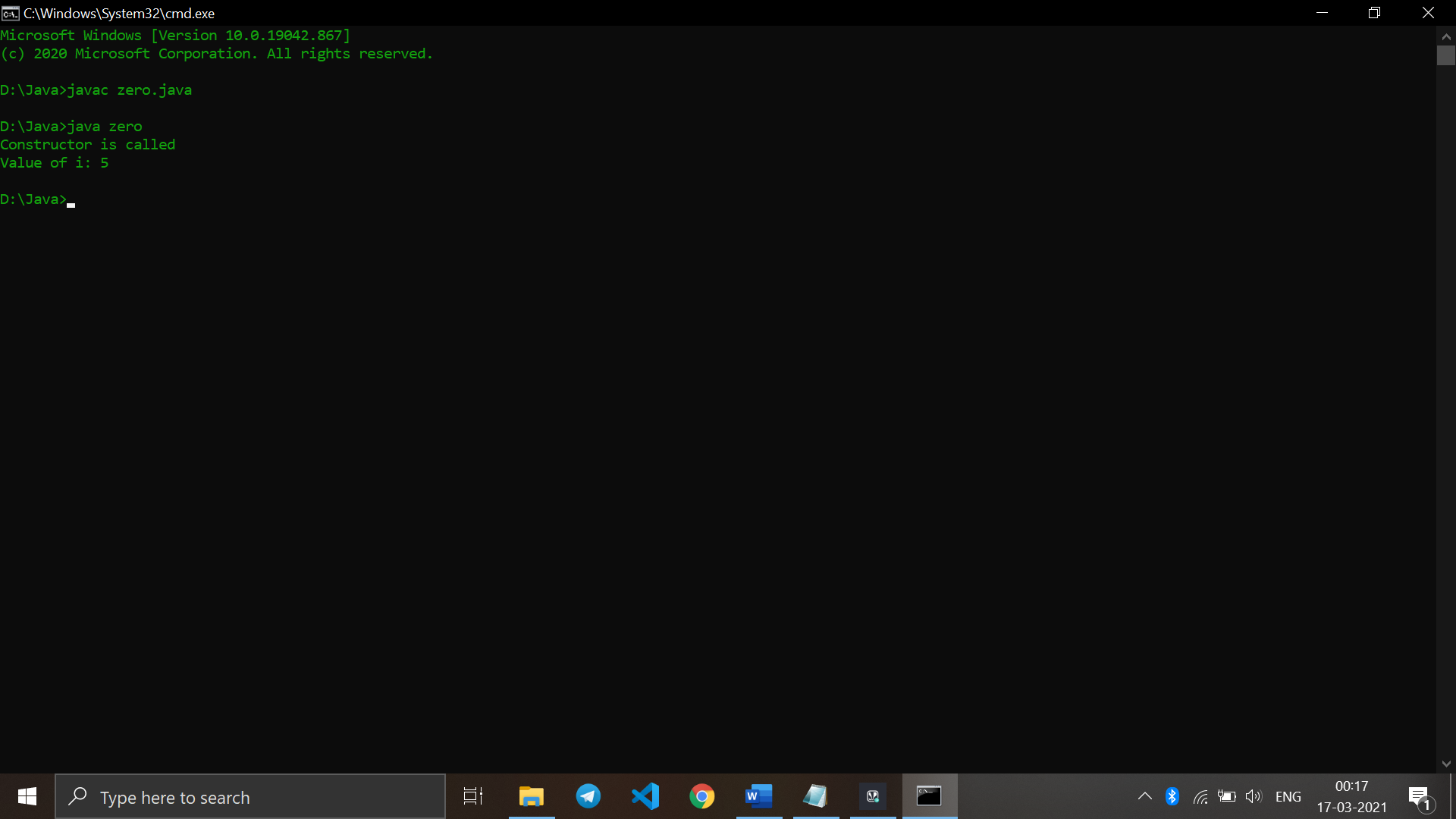
zero obj = new zero();

System.out.println("Value of i: " + obj.i);

}

}

Output :



Q25. Write a Java program to show parameterized constructor.

class para {

String languages;

para(String lang) {

languages = lang;

System.out.println(languages + " Programming Language");

}

public static void main(String[] args) {

para obj1 = new para("Java");

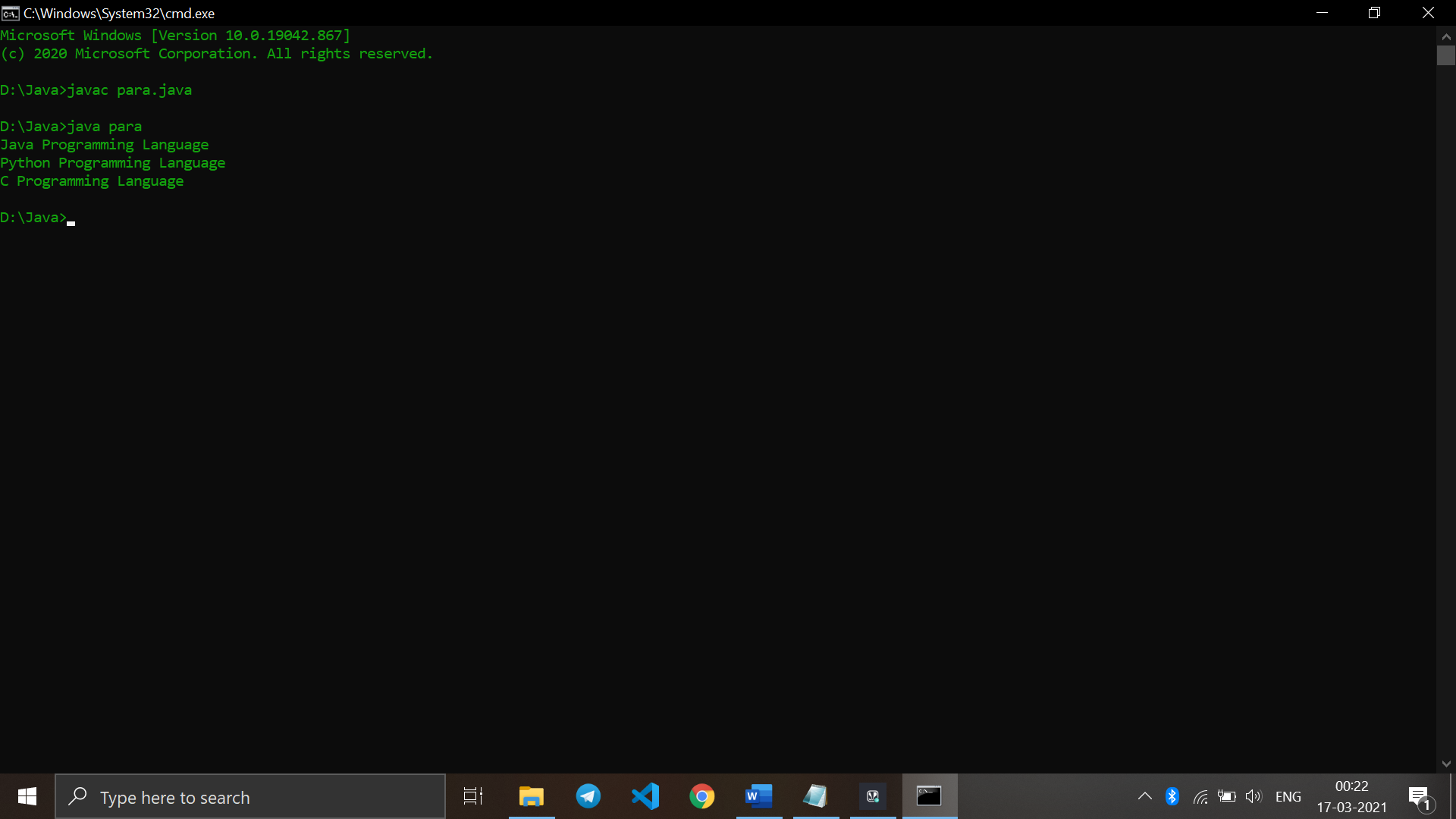
para obj2 = new para("Python");

para obj3 = new para("C");

}

}

Output :



Q26. Write a class, Grader, which has an instance variable, score, an appropriate constructor and appropriate methods. A method, letterGrade() that returns the letter grade as O/E/A/B/C/F.

Now write a demo class to test the Grader class by reading a score from the user, using it to create a Grader object after validating that the value is not negative and is not greater than 100. Finally, call the letterGrade() method to get and print the grade.

import java.util.Scanner;

class Grader {

public int s;

Grader(int s)

{

this.s=s;

}

public void letterGrade(){

if (this.s>90 && this.s<=100)

System.out.println("You Have Obtained : Grade O");

else if(this.s>80 && this.s<=90)

System.out.println("You Have Obtained : Grade E");

else if (this.s>70 && this.s<=80)

System.out.println("You Have Obtained : Grade A");

else if (this.s>60 && this.s<= 70)

System.out.println("You Have Obtained : Grade B");

else if (this.s>50 && this.s<= 60)

System.out.println("You Have Obtained : Grade C");

else

System.out.println("You Have Obtained : Grade F");

}

}

public class demo

{

public static void main(String[] args) {

int score;

Scanner sc = new Scanner(System.in);

System.out.print("Enter the marks obtained : ");

score = sc.nextInt();

if (score>=0 && score<=100){

Grader obj=new Grader(score);

obj.letterGrade();

}

else {

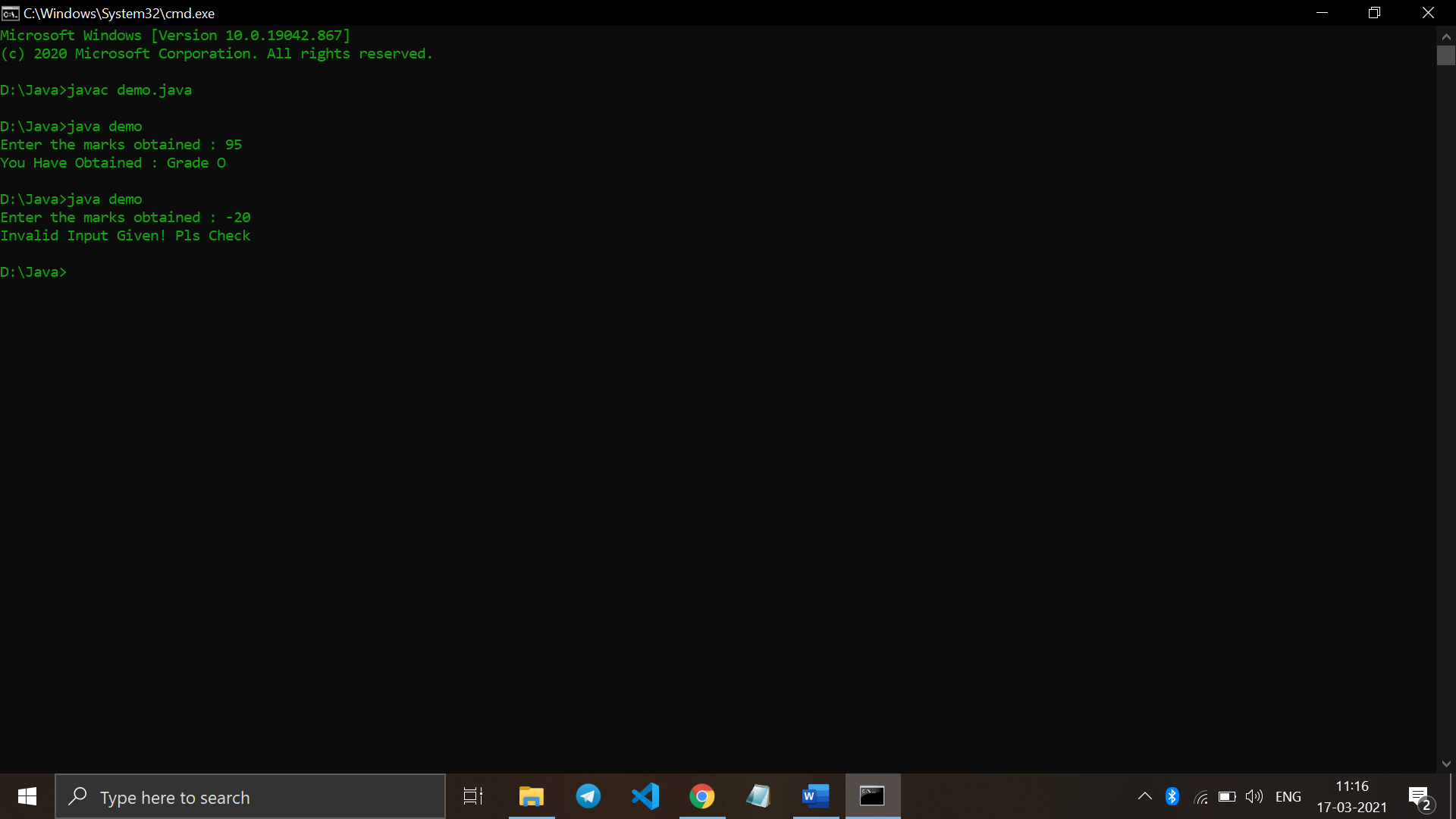
System.out.println("Invalid Input Given! Pls Check");

}

}

}

Output :



Q27. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission.

Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input”.

import java.util.\*;

class Commission {

public float sales;

Commission(float sales) {

this.sales = sales;

}

public double giveCommission() {

double result = this.sales - (this.sales \* 0.05);

return result;

}

}

public class demo1 {

// Main class to use commission class

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

float sales;

System.out.print("Enter the sales: ");

sales = sc.nextFloat();

if (sales < 0) {

System.out.println("Invalid Input");

} else {

Commission obj = new Commission(sales);

System.out.print("The commission = " + obj.giveCommission());

}

}

}

Output :

