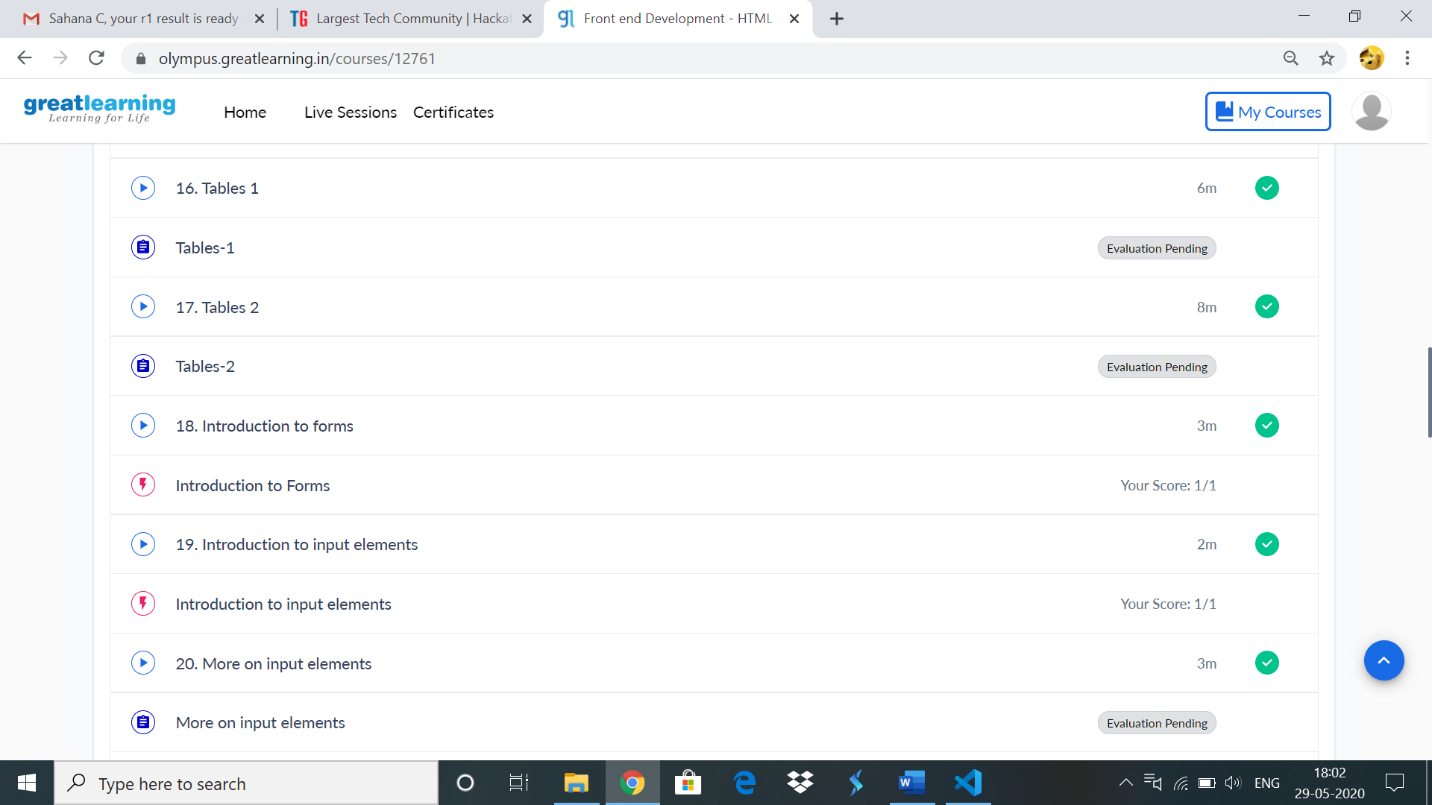
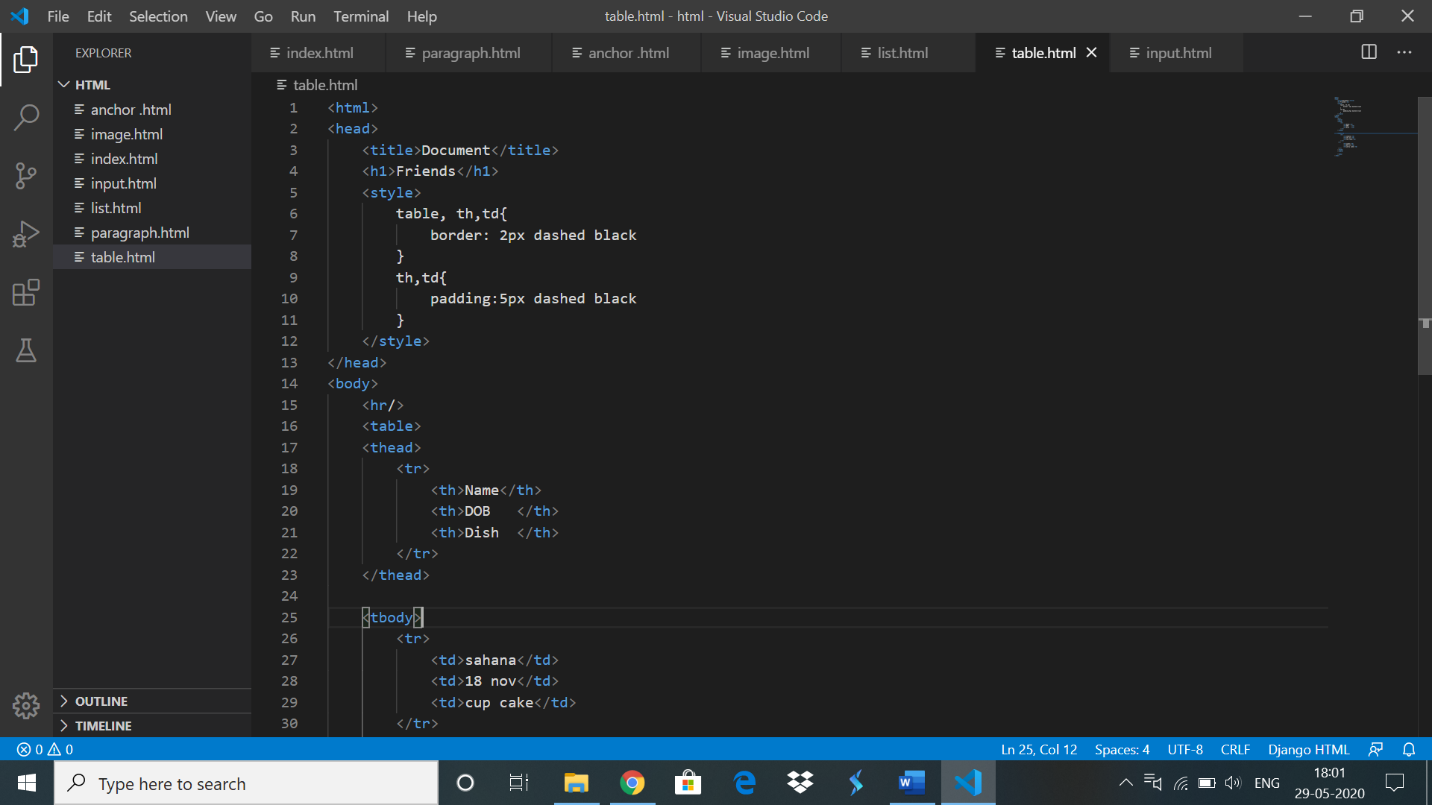
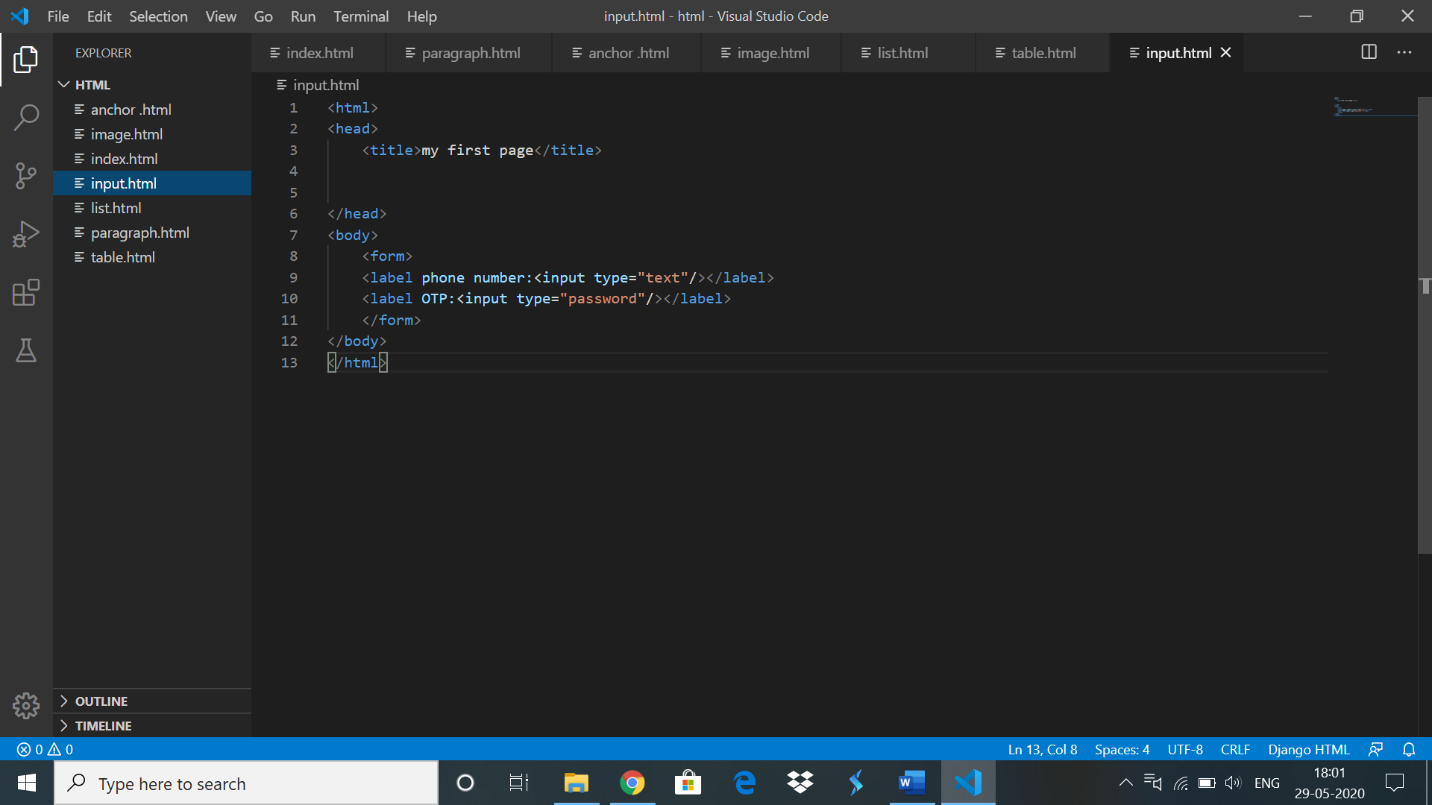
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **28-05-20** | | | | **Name:** | **Tanoj M** | |
| **Sem & Sec** | **VI A** | | | | **USN:** | **4AL16CS113** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **OR 2 IA Test** | | | | | |
| **Max. Marks** | | **30** | | **Score** | | **30** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **Front end development in html** | | | | | | |
| **Coding Challenges**   1. we are given 3 strings: str1, str2, and str3. Str3 is said to be a shuffle of str1 and str2 if it can be formed by interleaving the characters of str1 and str2 in a way that maintains the left to right ordering of the characters from each string. For example, given str1="abc" and str2="def", str3="dabecf" is a valid shuffle since it preserves the   prC22)c program to solve a system of linear congruences by applying the Chinese Remainder Theorem  P11)  1are  c | | | | | | | |
| **Certificate Provider** | | | **Great learning** | **Duration** | | | **6 days** |
| **Status:Completed** | | | | | | | |
| **Uploaded the report in Github** | | | | **Yes** | | | |
| **If yes Repository name** | | | | **https://github.com/Tanoj8296/DAILY-STATUS** | | | |
| **Uploaded the report in slack** | | | | **Yes** | | | |

**Online Certification Detail**







**Coding Challenge Details**

**1)**

**package** friday;

**import** java.util.\*;

**public** **class** findShortestSubString{

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

{

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

System.***out***.println("Enter the String1 : ");

String str1=s.next();

System.***out***.println("Enter the String2 : ");

String str2=s.next();

System.***out***.println("Enter the String1 : ");

String str3=s.next();

**int** j=0,k=0;

**for**(**int** i=0;i<str3.length();i++)

{

**if**(j<str1.length() && str3.charAt(i)==str1.charAt(j))

{

j++;

}

**else** **if**(k<str2.length() && str3.charAt(i)==str2.charAt(k))

{

k++;

}

**else**

{

**break**;

}

}

**if**(j==str1.length() && k==str2.length())

{

System.***out***.println("Valid Shuffle");

}

**else**

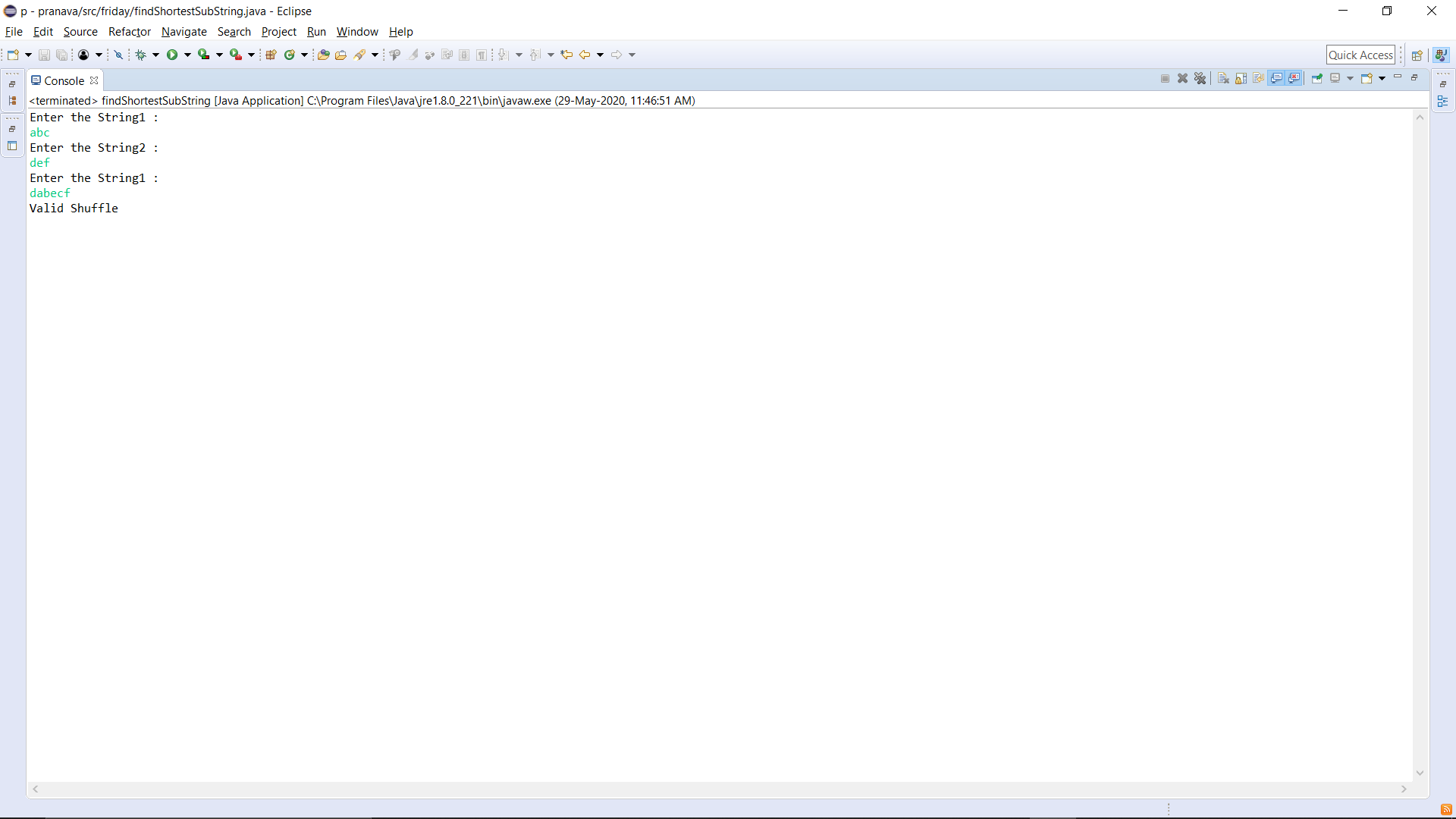
{

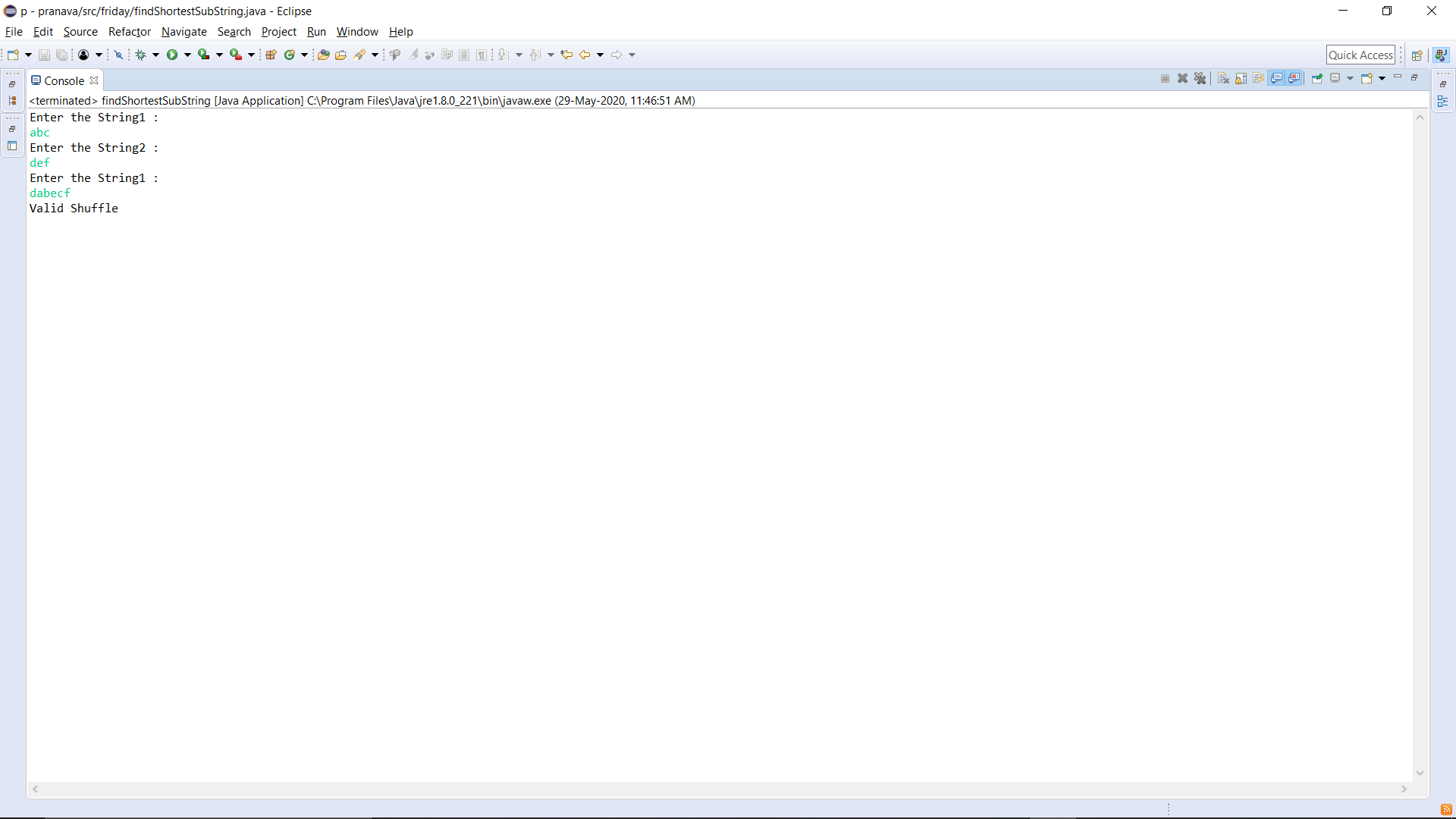
System.***out***.println("Invalid Shuffle");

}

}

}

**OUTPUT:**



2)

#include<stdio.h>

int findMinX(int num[], int rem[], int k)

{

int x = 1;

while (1)

{

Int j;

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

if (x%num[j] != rem[j])

break;

if (j == k)

return x;

x++;

}

return x;

}

int main(void)

{

int num[100];

int rem[100];

int k;

printf("Enter k value\n");

scanf("%d",&k);

printf("enter numbers\n");

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

scanf("%d",&num[i]);

printf("enter remainders\n");

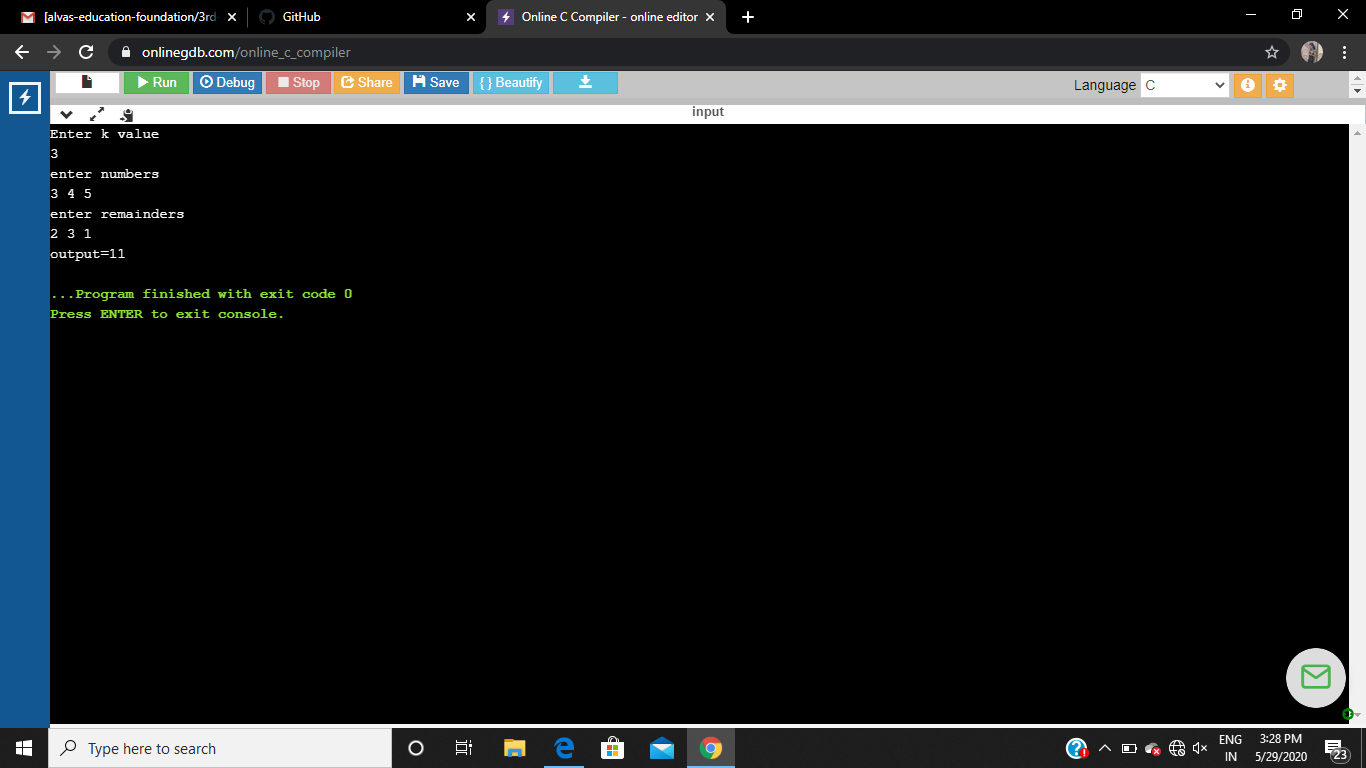
for(int i=0;i<k;i++)

scanf("%d",&rem[i]);

printf("output=%d",findMinX(num, rem, k));

return 0;

}



**OUTPUT**