Name Matching Problem (100 Marks)

The serious problem we are facing is the account opening fraud. In this growing type of fraud, criminals open accounts at financial institutions only to one day max out credit lines and cash advances before disappearing.

To avoid this, the institutions employ the screening process at the entry level. An exhaustive list of illegitimate and fraudulent account holders is maintained centrally which is considered as defaulting customers. The institution then will like to check every new account holder's detail with the defaulting list and will ensure that the new account holder name is not matching with defaulting list.

Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.

Please check the following test cases for more understanding.

Example 1
Modi Naitik
Mr. Naitik Modi
Match

It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.

In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.

Example 2
sumayya abou
SUMAYYABANU
Match

It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.

Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match

The above example specifies even if the subset matches from one string to the other, it is a match.

Example 4

SANKAR CHARAN DAS
s c das
Match

In this, the two strings are matched as their initials are matching. It is expected to match initials in any order. Hence Sankar Charan Das should also match with “c s das” or “das c s”.

Example 5
Mrs Meera Sakhare
Mira Sakhare
Match

This is example for phonetic matching. Two names sounding same but spelled differently should also match.

Example 6
Sureshkumar Periyal
Suresh Kumar Periyal
Match

In this example, two names are same, but they are bonded in different way.

Along with positive matching, there will be some test cases which are ‘No Match’. If two names are not getting matched with all the possible logic applied, then they are ‘No Match’. Please check the following examples for more clarity.

Example 7
Shri. Veer Prasad Lodha
Virchandra Prasad
No Match

Example 8
Radhika Sharma
Mrs. Sharma Kaveri
No Match

Input Format

The first line of input consists of Name 1.
The second line of input consist of Name 2.

Constraints

1<= Name 1, Name 2 <=50

Output Format

Print Match if the names are a match otherwise print No Match.

Sample TestCase 1

Input
Modi Naitik
Mr. Naitik Modi

Output
Match

Time Limit(X):

1.00 sec(s) for each input.

Memory Limit:

512 MB

Source Limit:

100 KB

Allowed Languages:

C, C++, C++11, C++14, C#, Java, Java 8, Kotlin, PHP, PHP 7, Python, Python 3, Perl, Ruby, Node Js, Scala, Clojure, Haskell, Lua, Erlang, Swift, VBnet, Js, Objc, Pascal, Go, F#, D, Groovy, Tcl, Ocaml, Smalltalk, Cobol, Racket, Bash, GNU Octave, Rust, Common LISP, R, Julia, Fortran, Ada, Prolog, Icon, Elixir, CoffeeScript, Brainfuck, Pypy, Lolcode, Nim, Picolisp, Pike, pypy3

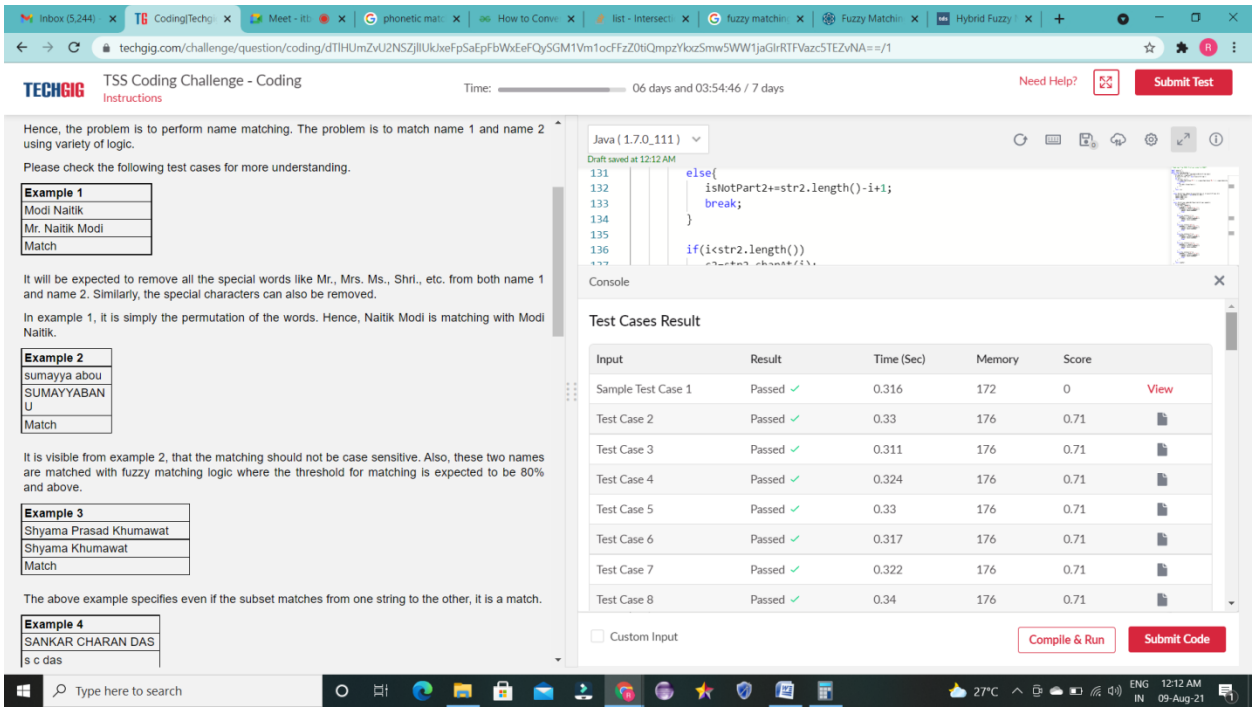
Soln:

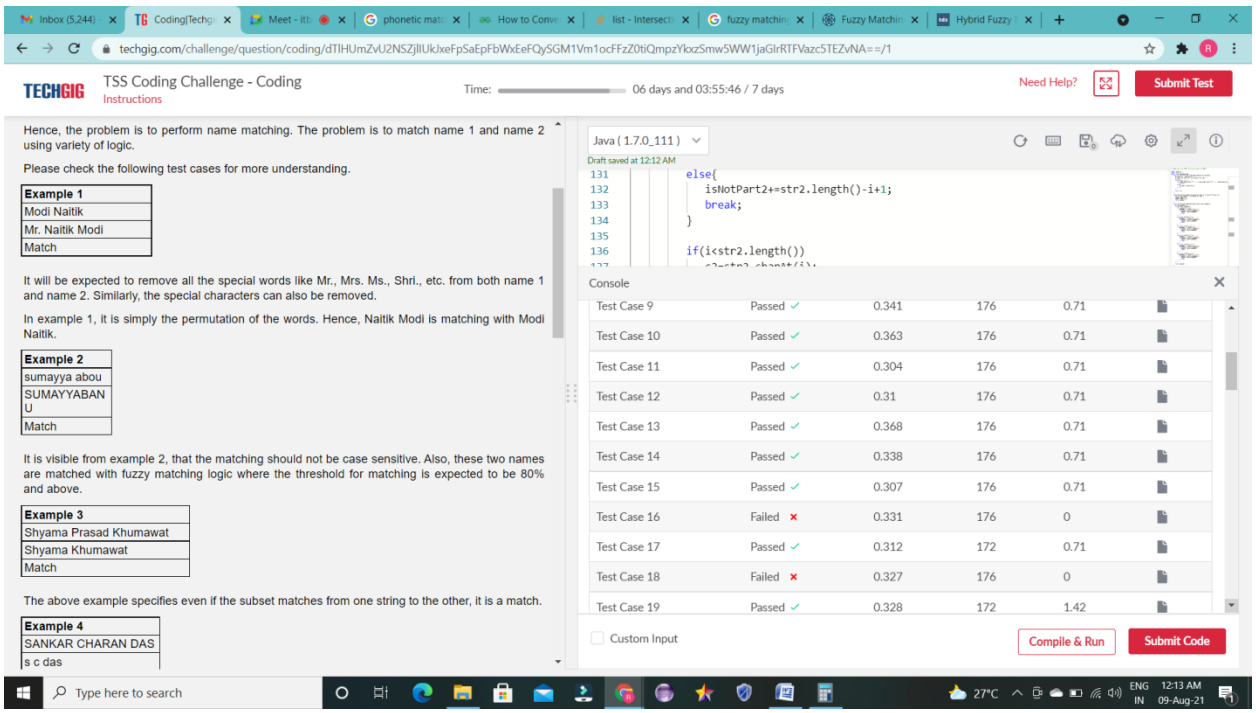
```
/* Read input from STDIN. Print your output to STDOUT*/

import java.io.*;
import java.util.*;
public class CandidateCode {
   static ArrayList<String> removeSpecialWords(String name){
      String[] arr = name.split(" ");
      ArrayList<String> res = new ArrayList<String>();
      int i=0;
      for(String s: arr){
         if(s.equalsIgnoreCase("Mr.") || s.equalsIgnoreCase("Ms.") || s.equalsIgnoreCase("Mrs.") ||
s.equalsIgnoreCase("Shri."))
            continue;
         else{
            res.add(s.toLowerCase());
            i++;
         }
      }
      return res;
   }

   static Set<String> isEqual(ArrayList<String> n1, ArrayList<String> n2){
      Set<String> nameSet = new HashSet<String>();
      nameSet.addAll(n1);
      nameSet.addAll(n2);
      return nameSet;
   }

   static Set<String> oneWordDiffSpell(Set<String> nameSet){
      String sameName="";
      for(String s: nameSet){
         if(s.contains("ee")){
            sameName = s.replace("ee","i");
            if(nameSet.contains(sameName)){
```

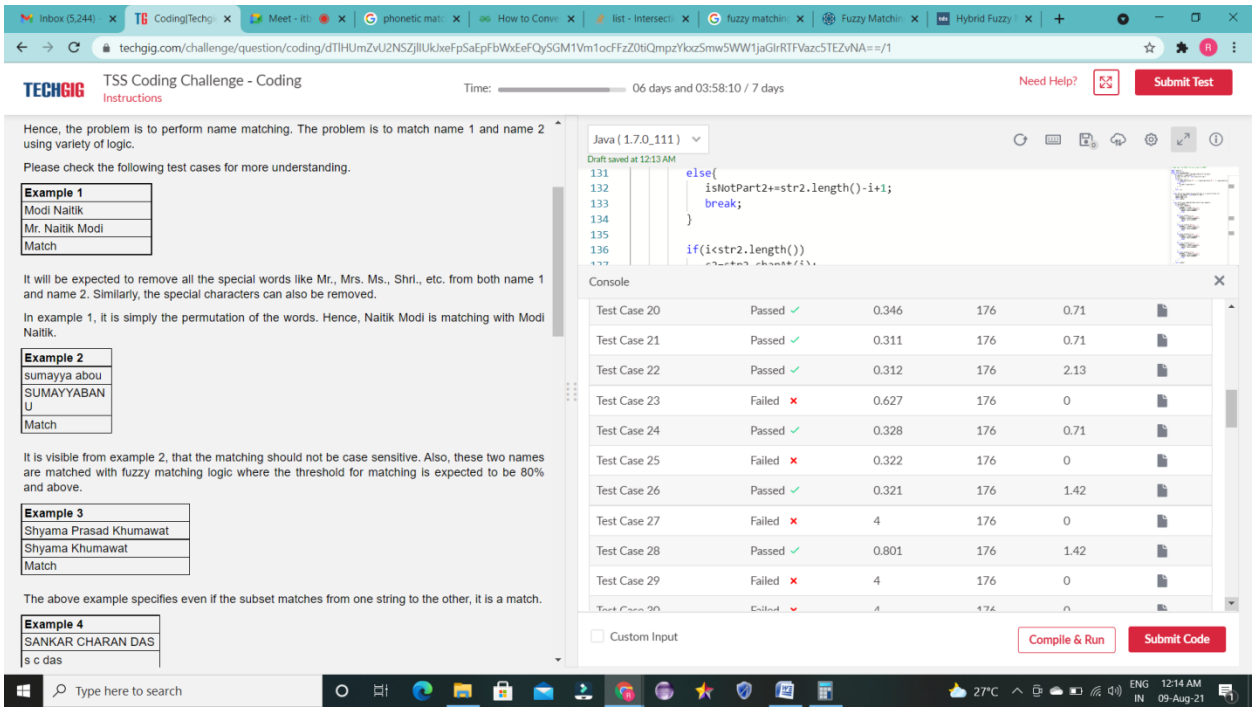
```
            nameSet.remove(sameName);
            break;
         }
      }
      else if(s.contains("ph")){
         sameName = s.replace("ph","f");
         if(nameSet.contains(sameName)){
            nameSet.remove(sameName);
            break;
         }
      }
      else if(s.contains("ie")){
         sameName = s.replace("ie","y");
         if(nameSet.contains(sameName)){
            nameSet.remove(sameName);
            break;
         }
      }
      else if(s.contains("ie")){
         sameName = s.replace("ie","ee");
         if(nameSet.contains(sameName)){
            nameSet.remove(sameName);
            break;
         }
      }
      else if(s.contains("ksh")){
         sameName = s.replace("ksh","x");
         if(nameSet.contains(sameName)){
            nameSet.remove(sameName);
            break;
         }
      }
   }
   return nameSet;
}

static Set<String> initialsPresent(Set<String> nameSet){
   int singleEle=0;
   while(true){
   for(String s : nameSet){
      if(s.length()>2){
         String str=""+s.charAt(0);
         if(nameSet.contains(str)){
            nameSet.remove(str);
            singleEle-=1;
            break;
         }
      }
      else{
         singleEle+=1;
      }
   }
   if(singleEle!=0){
      singleEle=0;
   }
   else if(singleEle==0)break;
   }
   return nameSet;
}

static boolean endProgram(Set<String> nameSet,ArrayList<String> nameArr1, ArrayList<String> nameArr2){
   if(nameSet.size() == nameArr1.size() || nameSet.size() == nameArr2.size()){
      return true;
   }
   return false;
}

static boolean diffBonded(ArrayList<String> nameArr1, ArrayList<String> nameArr2){
```

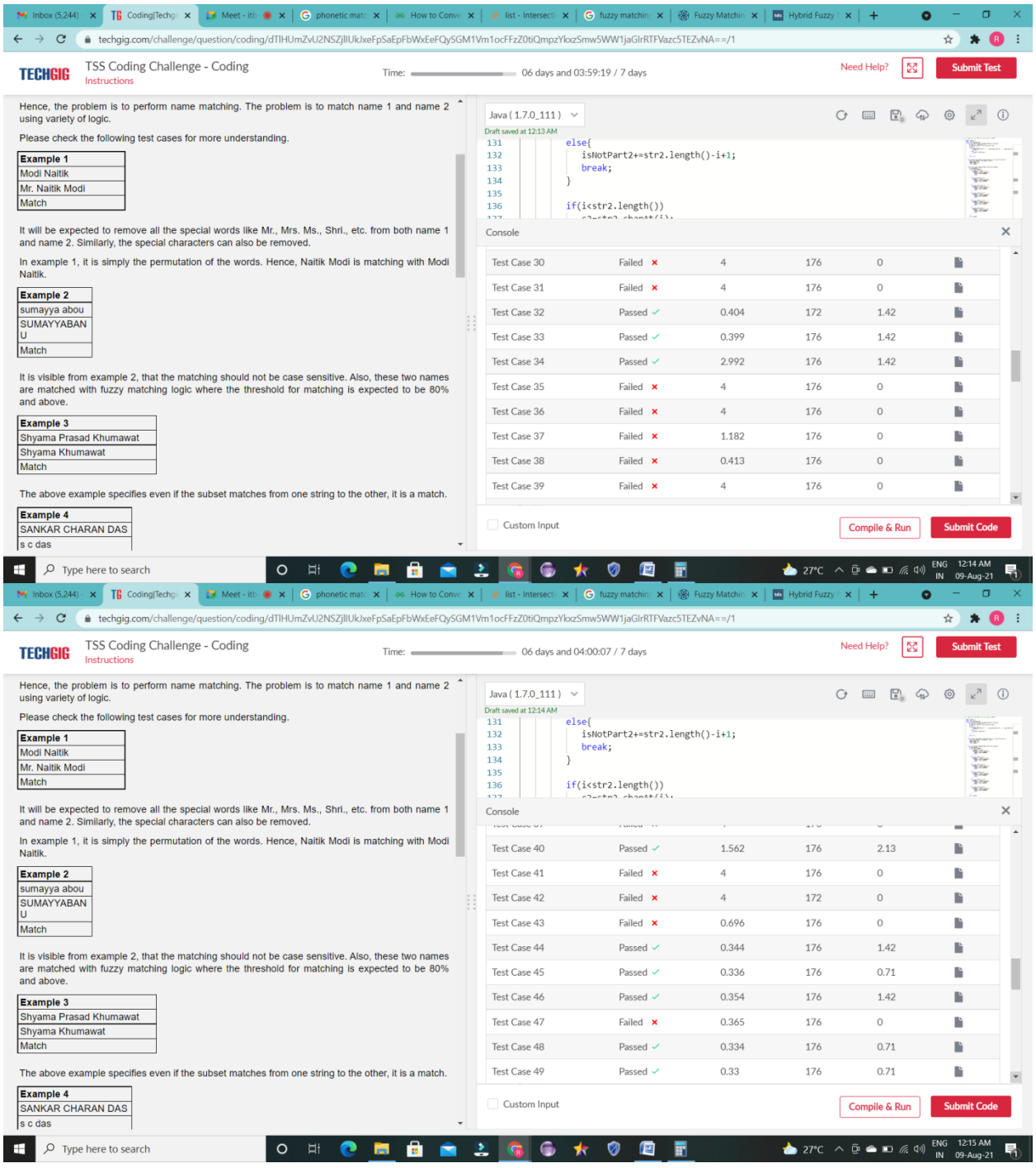
```
   String str1="";
   String str2="";
   for(String s: nameArr1){
      str1+=s;
   }
   for(String s: nameArr2){
      str2+=s;
   }
   if(str1.equals(str2))return true;
   //else return diffLogic(str1,str2);
   return false;
}

static boolean diffLogic(String str1,String str2){
   int len,i;
   int isPart=0,isNotPart1=0,isNotPart2=0;
   char c1,c2;
   if(str1.length()>=str2.length())len=str1.length();
   else len=str2.length();
   for(i=0;i<len;i++){
      if(i<str1.length())
         c1=str1.charAt(i);
      else{
         isNotPart2+=str2.length()-i+1;
         break;
      }

      if(i<str2.length())
         c2=str2.charAt(i);
      else{
         isNotPart1+=str1.length()-i+1;
         break;
      }

      if(c1==c2)isPart++;
      else{
         isNotPart1++;
         isNotPart2++;
      }
   }
   //i++;
   //isNotPart1=str1.length()-i;
   //isNotPart2=str2.length()-i;

   if(isPart>isNotPart1 || isPart>isNotPart2)
      return true;
   return false;
}

public static void main(String args[] ) throws Exception {
   String name1,name2;
   Scanner in = new Scanner(System.in);

   name1=in.nextLine();
   name2=in.nextLine();

   ArrayList nameArr1 = removeSpecialWords(name1);
   ArrayList nameArr2 = removeSpecialWords(name2);

   boolean endProgram=false;

   Set<String> nameSet = isEqual(nameArr1,nameArr2);

   if(endProgram(nameSet,nameArr1,nameArr2)){
      System.out.println("Match");
   }
   else {
      nameSet = initialsPresent(nameSet);
```

```
            if(endProgram(nameSet,nameArr1,nameArr2)){
               System.out.println("Match");
            }
            else{
               nameSet = oneWordDiffSpell(nameSet);
               if(endProgram(nameSet,nameArr1,nameArr2)){
                  System.out.println("Match");
               }
               else{
                  if(diffBonded(nameArr1,nameArr2)){
                     System.out.println("Match");
                  }
                  else{
                     if(diffLogic(name1.toLowerCase(),name2.toLowerCase())){
                        System.out.println("Match");
                     }
                     else System.out.println("No Match");
                  }
               }
            }
         }
      }
   }
```

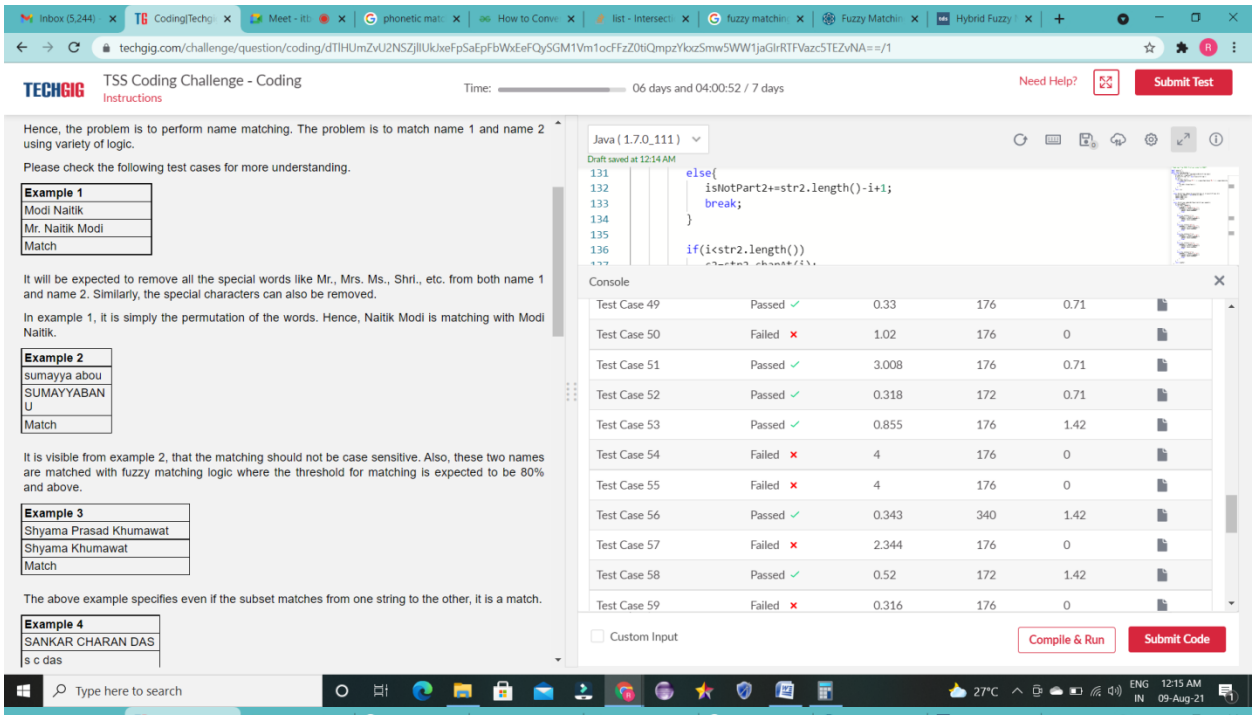

techgig.com/challenge/question/coding/dTlHUmZvU2NSZjllUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYlozSmw5WW1jaGlrRTFVazc5TEZvNA==/1
TECHGIG
TSS Coding Challenge - Coding
Instructions
Time: 06 days and 03:54:46 / 7 days
Need Help?
Submit Test
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABANU
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:12 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Cases Result
Input | Result | Time (Sec) | Memory | Score
Sample Test Case 1 | Passed | 0.316 | 172 | 0 | View
Test Case 2 | Passed | 0.33 | 176 | 0.71
Test Case 3 | Passed | 0.311 | 176 | 0.71
Test Case 4 | Passed | 0.324 | 176 | 0.71
Test Case 5 | Passed | 0.33 | 176 | 0.71
Test Case 6 | Passed | 0.317 | 176 | 0.71
Test Case 7 | Passed | 0.322 | 176 | 0.71
Test Case 8 | Passed | 0.34 | 176 | 0.71
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG IN
12:12 AM
09-Aug-21

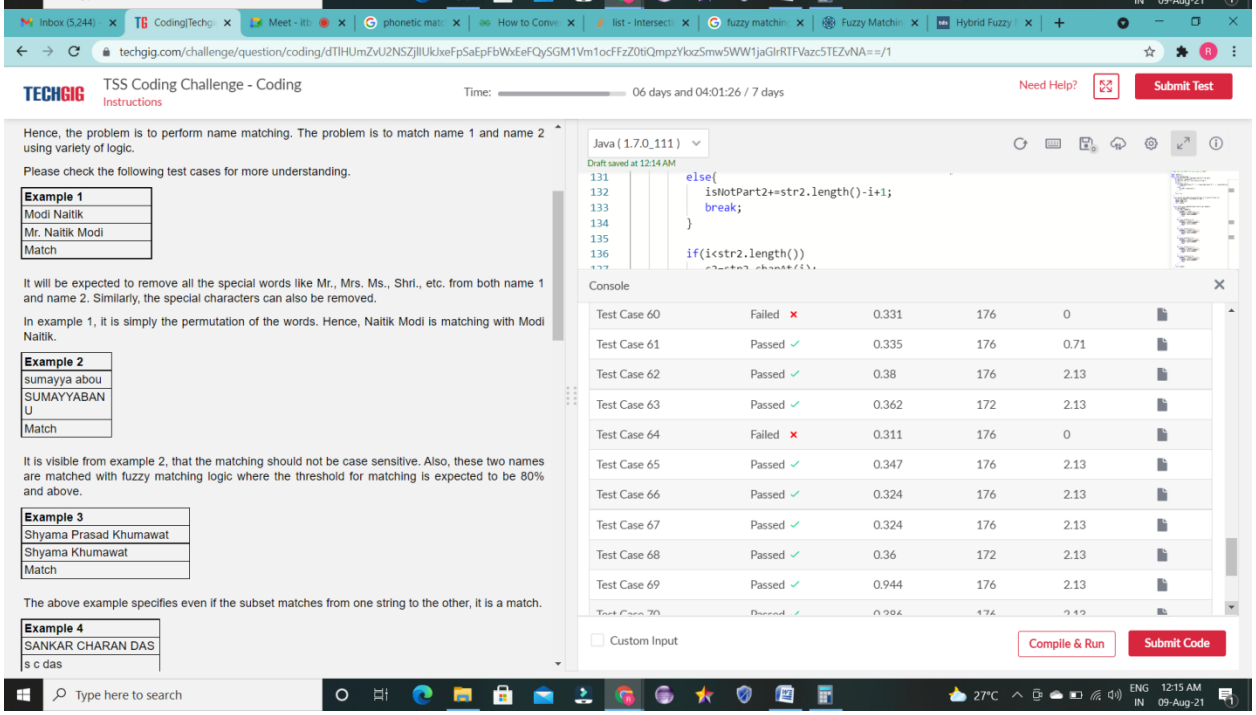

techgig.com/challenge/question/coding/dTlHUmZvU2NSZjllUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYlozSmw5WW1jaGlrRTFVazc5TEZvNA==/1
TECHGIG
TSS Coding Challenge - Coding
Instructions
Time: 06 days and 03:55:46 / 7 days
Need Help?
Submit Test
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABANU
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:12 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 9 | Passed | 0.341 | 176 | 0.71
Test Case 10 | Passed | 0.363 | 176 | 0.71
Test Case 11 | Passed | 0.304 | 176 | 0.71
Test Case 12 | Passed | 0.31 | 176 | 0.71
Test Case 13 | Passed | 0.368 | 176 | 0.71
Test Case 14 | Passed | 0.338 | 176 | 0.71
Test Case 15 | Passed | 0.307 | 176 | 0.71
Test Case 16 | Failed | 0.331 | 176 | 0
Test Case 17 | Passed | 0.312 | 172 | 0.71
Test Case 18 | Failed | 0.327 | 176 | 0
Test Case 19 | Passed | 0.328 | 172 | 1.42
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG IN
12:13 AM
09-Aug-21

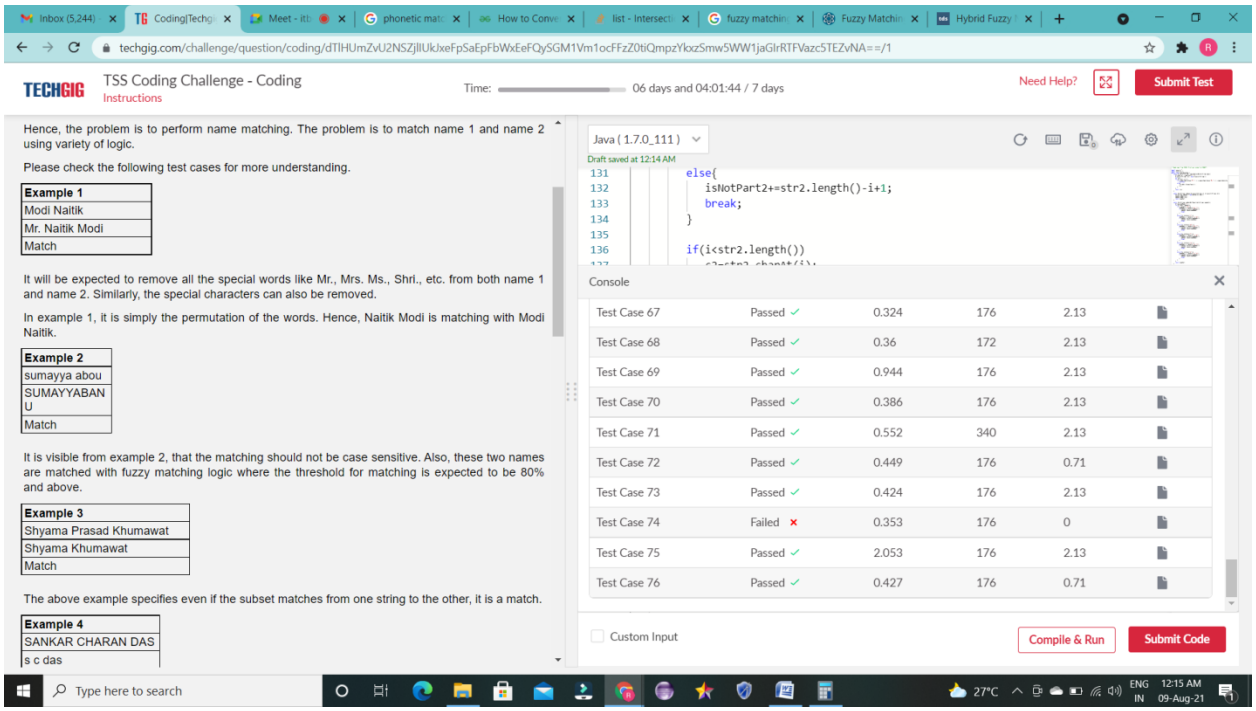

techgig.com/challenge/question/coding/dTlHUmZvU2NSZjllUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYkozSmw5WW1jaGlrRTFVazc5TEZvNA==/1
TECHGIG
TSS Coding Challenge - Coding
Instructions
Time:
06 days and 03:58:10 / 7 days
Need Help?
Submit Test
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABANU
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:13 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 20 Passed 0.346 176 0.71
Test Case 21 Passed 0.311 176 0.71
Test Case 22 Passed 0.312 176 2.13
Test Case 23 Failed 0.627 176 0
Test Case 24 Passed 0.328 176 0.71
Test Case 25 Failed 0.322 176 0
Test Case 26 Passed 0.321 176 1.42
Test Case 27 Failed 4 176 0
Test Case 28 Passed 0.801 176 1.42
Test Case 29 Failed 4 176 0
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG
IN
12:14 AM
09-Aug-21


TECHGIG
TSS Coding Challenge - Coding
Instructions
Time: 06 days and 03:59:19 / 7 days
Need Help?
Submit Test
techgig.com/challenge/question/coding/dTlHUmZvU2NSZjlIUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYkxzSmw5WW1jaGlrRTFVazc5TEZvNA==/1
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABAN U
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:13 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 30 Failed 4 176 0
Test Case 31 Failed 4 176 0
Test Case 32 Passed 0.404 172 1.42
Test Case 33 Passed 0.399 176 1.42
Test Case 34 Passed 2.992 176 1.42
Test Case 35 Failed 4 176 0
Test Case 36 Failed 4 176 0
Test Case 37 Failed 1.182 176 0
Test Case 38 Failed 0.413 176 0
Test Case 39 Failed 4 176 0
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG IN
12:14 AM
09-Aug-21

TECHGIG
TSS Coding Challenge - Coding
Instructions
Time: 06 days and 04:00:07 / 7 days
Need Help?
Submit Test
techgig.com/challenge/question/coding/dTlHUmZvU2NSZjlIUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYkxzSmw5WW1jaGlrRTFVazc5TEZvNA==/1
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABAN U
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:14 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 40 Passed 1.562 176 2.13
Test Case 41 Failed 4 176 0
Test Case 42 Failed 4 172 0
Test Case 43 Failed 0.696 176 0
Test Case 44 Passed 0.344 176 1.42
Test Case 45 Passed 0.336 176 0.71
Test Case 46 Passed 0.354 176 1.42
Test Case 47 Failed 0.365 176 0
Test Case 48 Passed 0.334 176 0.71
Test Case 49 Passed 0.33 176 0.71
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG IN
12:15 AM
09-Aug-21


techgig.com/challenge/question/coding/dTlHUmZvU2NSZjllUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYkxzSmw5WW1jaGlrRTFVazc5TEZvNA==/1
TECHGIG
TSS Coding Challenge - Coding
Instructions
Time: 06 days and 04:00:52 / 7 days
Need Help?
Submit Test
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABANU
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:14 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 49 Passed 0.33 176 0.71
Test Case 50 Failed 1.02 176 0
Test Case 51 Passed 3.008 176 0.71
Test Case 52 Passed 0.318 172 0.71
Test Case 53 Passed 0.855 176 1.42
Test Case 54 Failed 4 176 0
Test Case 55 Failed 4 176 0
Test Case 56 Passed 0.343 340 1.42
Test Case 57 Failed 2.344 176 0
Test Case 58 Passed 0.52 172 1.42
Test Case 59 Failed 0.316 176 0
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG IN
12:15 AM 09-Aug-21


techgig.com/challenge/question/coding/dTlHUmZvU2NSZjllUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYkxzSmw5WW1jaGlrRTFVazc5TEZvNA==/1
TECHGIG
TSS Coding Challenge - Coding
Instructions
Time: 06 days and 04:01:26 / 7 days
Need Help?
Submit Test
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABANU
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:14 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 60 Failed 0.331 176 0
Test Case 61 Passed 0.335 176 0.71
Test Case 62 Passed 0.38 176 2.13
Test Case 63 Passed 0.362 172 2.13
Test Case 64 Failed 0.311 176 0
Test Case 65 Passed 0.347 176 2.13
Test Case 66 Passed 0.324 176 2.13
Test Case 67 Passed 0.324 176 2.13
Test Case 68 Passed 0.36 172 2.13
Test Case 69 Passed 0.944 176 2.13
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG IN
12:15 AM 09-Aug-21


techgig.com/challenge/question/coding/dTlHUmZvU2NSZjlUkJxeFpSaEpFbWxEeFQySGM1Vm1ocFFzZ0tiQmpzYkozSmw5WW1jaGlrRTFVazc5TEZvNA==/1
TECHGIG
TSS Coding Challenge - Coding
Instructions
Time:
06 days and 04:01:44 / 7 days
Need Help?
Submit Test
Hence, the problem is to perform name matching. The problem is to match name 1 and name 2 using variety of logic.
Please check the following test cases for more understanding.
Example 1
Modi Naitik
Mr. Naitik Modi
Match
It will be expected to remove all the special words like Mr., Mrs. Ms., Shri., etc. from both name 1 and name 2. Similarly, the special characters can also be removed.
In example 1, it is simply the permutation of the words. Hence, Naitik Modi is matching with Modi Naitik.
Example 2
sumayya abou
SUMAYYABANU
Match
It is visible from example 2, that the matching should not be case sensitive. Also, these two names are matched with fuzzy matching logic where the threshold for matching is expected to be 80% and above.
Example 3
Shyama Prasad Khumawat
Shyama Khumawat
Match
The above example specifies even if the subset matches from one string to the other, it is a match.
Example 4
SANKAR CHARAN DAS
s c das
Java (1.7.0_111)
Draft saved at 12:14 AM
131 else{
132 isNotPart2+=str2.length()-i+1;
133 break;
134 }
135
136 if(i<str2.length())
Console
Test Case 67 Passed 0.324 176 2.13
Test Case 68 Passed 0.36 172 2.13
Test Case 69 Passed 0.944 176 2.13
Test Case 70 Passed 0.386 176 2.13
Test Case 71 Passed 0.552 340 2.13
Test Case 72 Passed 0.449 176 0.71
Test Case 73 Passed 0.424 176 2.13
Test Case 74 Failed 0.353 176 0
Test Case 75 Passed 2.053 176 2.13
Test Case 76 Passed 0.427 176 0.71
Custom Input
Compile & Run
Submit Code
Type here to search
27°C
ENG
IN
12:15 AM
09-Aug-21