DP-S016 (E043)

Solved Challenges **0/1**

Back To Challenges List





Minimum Edit Distance Two Strings

Solved By 476 Users ID:11156

The program must accept two string values S1 and S2 as the input. The program must print the minimum cost required to convert the string S1 to S2 as the output. The cost of insertion, deletion and substitution of any character in the string S1 will be 1.

Boundary Condition(s):

1 <= Length of S1, S2 <= 1000

Input Format:

The first line contains S1.

The second line contains S2.

Output Format:

The first line contains the minimum cost required to convert the string S1 to S2.

Example Input/Output 1:

Input:

hello

hail

Output:

3

Explanation: Here S1 = hello and S2 = hail The minimum cost required to convert the string hello to hail is 3. The 3 operations are given below. 1 - a is substituted in place of e. Now the string S1 becomes hallo. 2 - i is substituted in place of I (first occurring I). Now the string S1 becomes hailo. 3 - o is deleted. Now the string S1 becomes hail.

Example Input/Output 2:

Input:

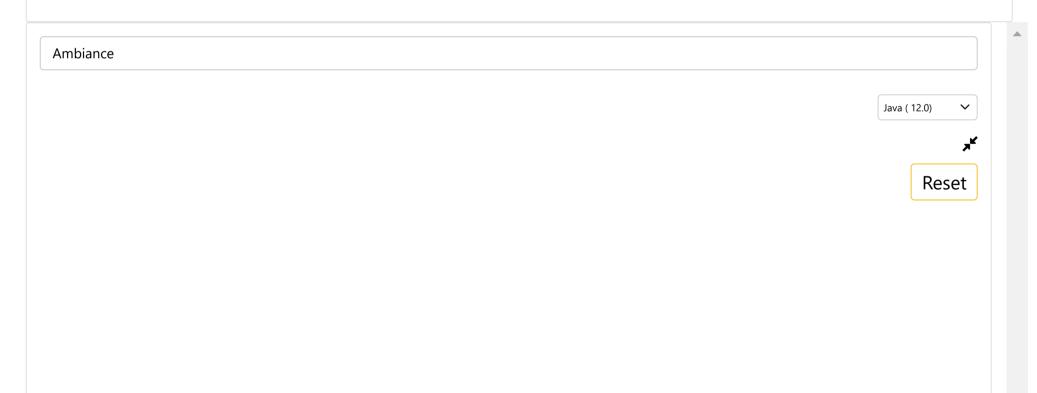
intrinsic

intrusive

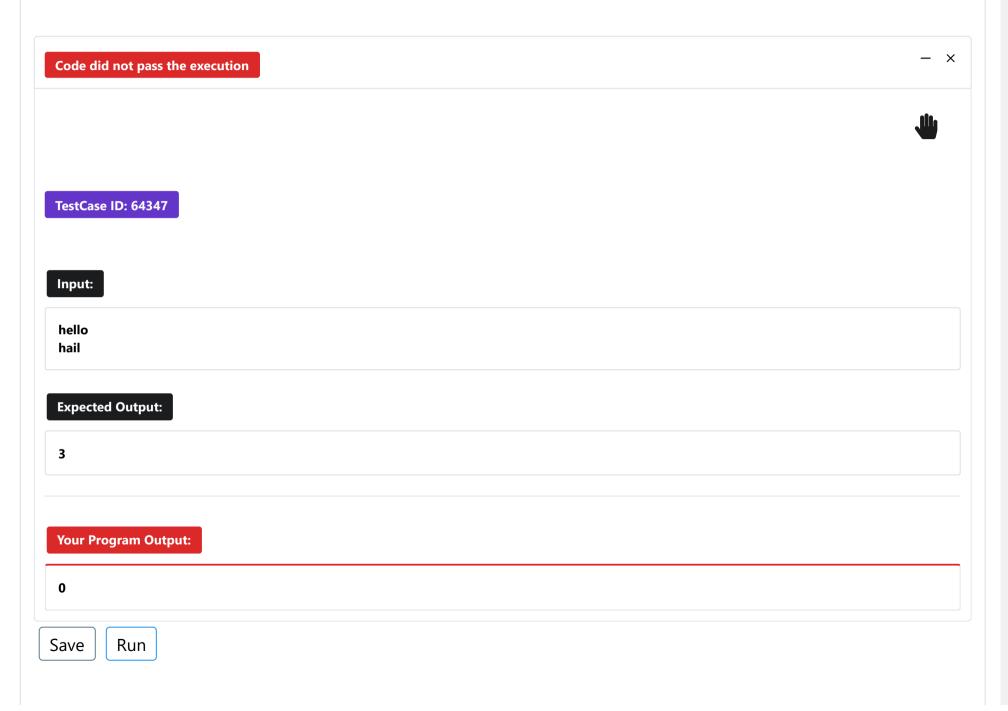
Output:

4

Max Execution Time Limit: 500 millisecs



```
1 import java.util.*;
   public class Hello {
 3
        public static void main(String[] args) {
 4
            Scanner sc = new Scanner(System.in);
            String s1 = sc.next();
 6
            String s2 = sc.next();
 7
 8
 9
            int matrix[][] = new int[s2.length()+1][s1.length()+1];
10
11
            for(int col = 0;col<=s1.length();col++)</pre>
12
                matrix[0][col]=col;
13
            for(int row = 0;row<=s2.length();row++)</pre>
                matrix[row][0]=row;
14
15
16
            for(int row=1;row<=s2.length();row++)</pre>
17
                 for(int col=1;col<=s1.length();col++)</pre>
18
19
20
                     if(s1.charAt(col-1) == s2.charAt(row-1))
21
22
                         matrix[row][col] = matrix[row-1][col-1];
23
24
                     else
25
26
                         int left = matrix[row][col-1];
27
                         int top = matrix[row-1][col];
28
                         int topleft = matrix[row-1][col-1];
                         matrix[row][col] = Math.min(topleft, Math.min(left,top)) + 1;
29
30
31
32
33
34
            System.out.println(matrix[s2.length()][s1.length()]);
35
36
        }
37
   }
```



Run with a custom test case (Input/Output)