LP_Practice_EncodingThreeStrings

Ramya.V | 12 Feb 2023

Test Name::

LP_Practice_EncodingThreeStrings

Finish State: Normal | Test Taken on: February 12, 2023 07:38:47 PM IST

R

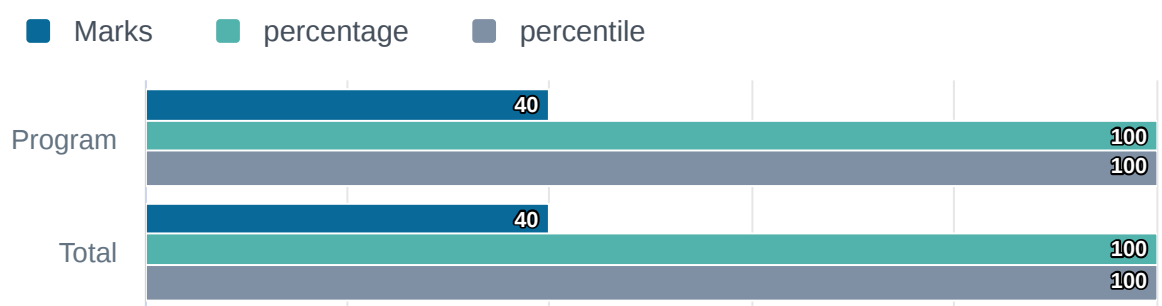
Ramya.V

ramya.v.2020.cse@ritchennai.edu.in

Overall Summary

40	Marks Scored	out of 40
100 %	100 percentile	out of 44417 Test Takers
5m 45s	Time taken	of 1hr 20mins

Marks Scored



Attempt Summary

Distribution of questions attempted in a total of 1 question(s).

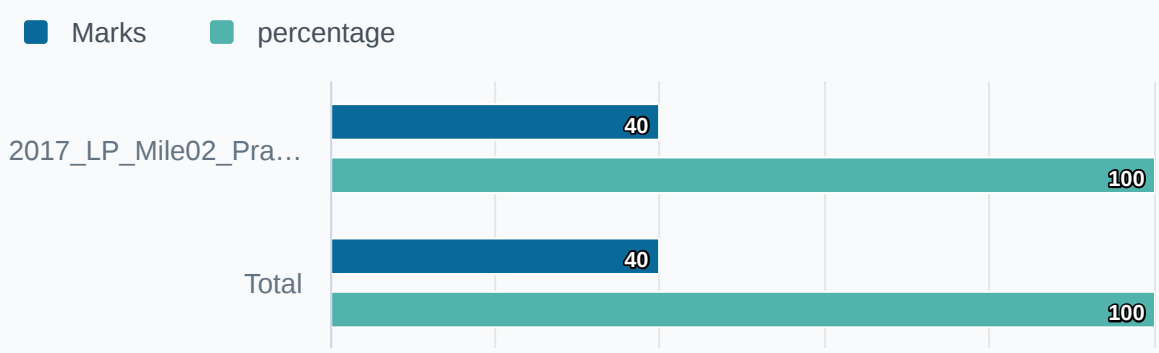
This shows the correctness of questions attempted by the test taker

Correct	1 Ques	40/40 Marks
Incorrect	0 Ques	0/0 Marks
Partially Correct	0 Ques	0/0 Marks
Not Attempted	0 Ques	0/0 Marks

Section-Wise Details

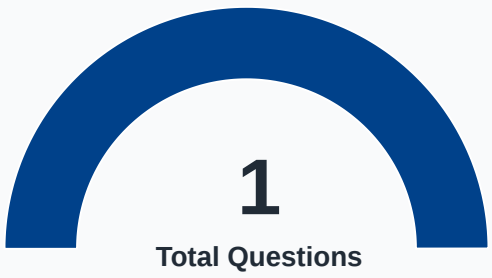
Section 1	question(s)	Time taken	Marks Scored
Program	1 Q.	5m 45s (Untimed)	40 / 40

Marks Scored



Attempt Summary

Distribution of questions attempted in a total of 1 question(s).



This shows the correctness of questions attempted by the test taker

Correct	1 Ques	40/40 Marks

Test Log

12th Feb 2023

07:32 PM	Started the test with Program
07:34 PM	Away from test window
07:37 PM	Away from test window
07:37 PM	Away from test window
07:38 PM	Away from test window
07:38 PM	Away from test window
07:38 PM	Finished the test

About the Report

This Report is generated electronically on the basis of the inputs received from the assessment takers. This Report including the AI flags that are generated in case of availing of proctoring services, should not be solely used/relied on for making any business, selection, entrance, or employment-related decisions. Mettl accepts no liability from the use of or any action taken or refrained from or for any and all business decisions taken as a result of or reliance upon anything, including, without limitation, information, advice, or AI flags contained in this Report or sources of information used or referred to in this Report.

Mercer | mettl | Ramya.V
LP_Practice_EncodingThreeStrings / Saved: 60 seconds ago
Test Time: 01:15:46 | Finish Test

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Encoding Three Strings: Anand was assigned the task of coming up with an encoding mechanism for any given three strings. He has come up with the below plan.

STEP ONE: Given any three strings, break each string into 3 parts each.
For example – If the three strings are as below -
Input1= "John"
Input2= "Johny"
Input3= "Janardhan"
"John" should be split into "J", "oh", "n" as the FRONT, MIDDLE and END parts respectively.
"Johny" should be split into "Jo", "h", "ny" as the FRONT, MIDDLE and END parts respectively.
"Janardhan" should be split into "Jan", "ard", "han" as the FRONT, MIDDLE and END parts respectively.
i.e. if the no. of characters in the string are in multiples of 3, then each split-part will contain equal no. of characters, as seen in the example of "Janardhan"
If the no. of characters in the string are NOT in multiples of 3, and if there is one character more than multiple of 3, then the middle part will get the extra character, as seen in the example of "John"
If the no. of characters in the string are NOT in multiples of 3, and if there are two characters more than multiple of 3, then the FRONT and END parts will get one extra character each, as seen in the example of "Johny"

STEP TWO: Concatenate (join) the FRONT, MIDDLE and END parts of the strings as per the below specified concatenation-rule to form three Output strings.
Output1 = FRONT part of Input1 + FRONT part of Input2 + FRONT part of Input3
Output2 = MIDDLE part of Input1 + MIDDLE part of Input2 + MIDDLE part of Input3

JAVA7 | Compiler: Java - 1.7

```
1  import java.io.*;
2  import  java.util.*;
3
4  // Read only region start
5  class UserMainCode
6  {
7
8      public class Result{
9          public final String output1;
10         public final String output2;
11         public final String output3;
12
13         public Result(String out1, String out2, String out3){
14             output1 = out1;
15             output2 = out2;
16             output3 = out3;
17         }
18     }
19
20     public Result encodeThreeStrings(String input1,String input2,String input3){
21         // Read only region end
22       String frnt1="",mid1="",end1="";
23     String frnt2="",mid2="",end2="";
24     String frnt3="",mid3="",end3="";
25     String output1="",output2="",output3="";
26     int len1=input1.length();
```

Use Custom Input | Compile and Test | Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Encoding Three Strings: Anand was assigned the task of coming up with an encoding mechanism for any given three strings. He has come up with the below plan.

STEP ONE: Given any three strings, break each string into 3 parts each.
For example – If the three strings are as below -
Input1= "John"
Input2= "Johny"
Input3= "Janardhan"
"John" should be split into "J", "oh", "n" as the FRONT, MIDDLE and END parts respectively.
"Johny" should be split into "Jo", "h", "ny" as the FRONT, MIDDLE and END parts respectively.
"Janardhan" should be split into "Jan", "ard", "han" as the FRONT, MIDDLE and END parts respectively.
i.e. if the no. of characters in the string are in multiples of 3, then each split-part will contain equal no. of characters, as seen in the example of "Janardhan"
If the no. of characters in the string are NOT in multiples of 3, and if there is one character more than multiple of 3, then the middle part will get the extra character, as seen in the example of "John"
If the no. of characters in the string are NOT in multiples of 3, and if there are two characters more than multiple of 3, then the FRONT and END parts will get one extra character each, as seen in the example of "Johny"

STEP TWO: Concatenate (join) the FRONT, MIDDLE and END parts of the strings as per the below specified concatenation-rule to form three Output strings.
Output1 = FRONT part of Input1 + FRONT part of Input2 + FRONT part of Input3
Output2 = MIDDLE part of Input1 + MIDDLE part of Input2 + MIDDLE part of Input3

JAVA7 | Compiler: Java - 1.7

```
28      int len3=input3.length();
29    if(len1==input1.length()){
30    if(len1%3==0)
31    {
32     frnt1=input1.substring(0, (len1/3));
33     mid1=input1.substring((len1/3), (2*(len1/3)));
34     end1=input1.substring(2*(len1/3));
35    }
36    else if((len1-1)%3==0)
37    {
38     frnt1=input1.substring(0, (len1/3));
39     mid1=input1.substring((len1/3), ((2*(len1/3))+1));
40     end1=input1.substring(((2*(len1/3))+1));
41    }
42    else if((len1-2)%3==0)
43    {
44     frnt1=input1.substring(0, ((len1/3)+1));
45     mid1=input1.substring(((len1/3)+1), ((2*(len1/3))+1));
46     end1=input1.substring(((2*(len1/3))+1));
47    }
48    }
49     if(len2==input2.length()){
50    if(len2%3==0)
51    {
52     frnt2=input2.substring(0, (len2/3));
53     mid2=input2.substring((len2/3), (2*(len2/3)));
```

Use Custom Input

Compile and Test | Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Encoding Three Strings: Anand was assigned the task of coming up with an encoding mechanism for any given three strings. He has come up with the below plan.

STEP ONE: Given any three strings, break each string into 3 parts each.
For example – If the three strings are as below -
Input1= "John"
Input2= "Johny"
Input3= "Janardhan"
"John" should be split into "J", "oh", "n" as the FRONT, MIDDLE and END parts respectively.
"Johny" should be split into "Jo", "h", "ny" as the FRONT, MIDDLE and END parts respectively.
"Janardhan" should be split into "Jan", "ard", "han" as the FRONT, MIDDLE and END parts respectively.
i.e. if the no. of characters in the string are in multiples of 3, then each split-part will contain equal no. of characters, as seen in the example of "Janardhan"
If the no. of characters in the string are NOT in multiples of 3, and if there is one character more than multiple of 3, then the middle part will get the extra character, as seen in the example of "John"
If the no. of characters in the string are NOT in multiples of 3, and if there are two characters more than multiple of 3, then the FRONT and END parts will get one extra character each, as seen in the example of "Johny"

STEP TWO: Concatenate (join) the FRONT, MIDDLE and END parts of the strings as per the below specified concatenation-rule to form three Output strings.
Output1 = FRONT part of Input1 + FRONT part of Input2 + FRONT part of Input3
Output2 = MIDDLE part of Input1 + MIDDLE part of Input2 + MIDDLE part of Input3

JAVA7 | Compiler: Java - 1.7

```
91   output3=end1+end2+end3;
92   System.out.println(output3);
93   output3=changeCase(output3);
94   Result rs=new Result(output1,output2,output3);
95   return rs;
96     }
97  public static String changeCase(String str)
98  {
99   StringBuffer newS = new StringBuffer(str);
100  for(int i=0;i<str.length();i++)
101  {
102   Character c=str.charAt(i);
103   if(Character.isLowerCase(c))
104   {
105    newS.replace(i, i+1, Character.toUpperCase(c)+"");
106   }
107   else
108   {
109    newS.replace(i, i+1, Character.toLowerCase(c)+"");
110   }
111  }
112  str=newS.toString();
113  return str;
114 }
115 }
116
```

Use Custom Input

Compile and Test | Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Encoding Three Strings: Anand was assigned the task of coming up with an encoding mechanism for any given three strings. He has come up with the below plan.

STEP ONE: Given any three strings, break each string into 3 parts each.
For example – If the three strings are as below -
Input1= "John"
Input2= "Johny"
Input3= "Janardhan"
"John" should be split into "J", "oh", "n" as the FRONT, MIDDLE and END parts respectively.
"Johny" should be split into "Jo", "h", "ny" as the FRONT, MIDDLE and END parts respectively.
"Janardhan" should be split into "Jan", "ard", "han" as the FRONT, MIDDLE and END parts respectively.
i.e. if the no. of characters in the string are in multiples of 3, then each split-part will contain equal no. of characters, as seen in the example of "Janardhan"
If the no. of characters in the string are NOT in multiples of 3, and if there is one character more than multiple of 3, then the middle part will get the extra character, as seen in the example of "John"
If the no. of characters in the string are NOT in multiples of 3, and if there are two characters more than multiple of 3, then the FRONT and END parts will get one extra character each, as seen in the example of "Johny"

STEP TWO: Concatenate (join) the FRONT, MIDDLE and END parts of the strings as per the below specified concatenation-rule to form three Output strings.
Output1 = FRONT part of Input1 + FRONT part of Input2 + FRONT part of Input3
Output2 = MIDDLE part of Input1 + MIDDLE part of Input2 + MIDDLE part of Input3

default

CODE EXECUTION DETAILS

Time: 130 ms

Memory: 103812 kb

TEST CASE INFORMATION

Input

John,Johny,Janardhan

Expected Output

JJoJan,ohhard,NNYHAN

Actual Output

JJoJan,ohhard,NNYHAN

CONSOLE OUTPUT

nnyhan

STANDARD ERROR/WARNING

None