LP_Practice_IdentifyPossibleWords

Ramya.V | 12 Feb 2023

Test Name::

LP_Practice_IdentifyPossibleWords

Finish State: Normal | Test Taken on: February 12, 2023 07:26:36 PM IST

R

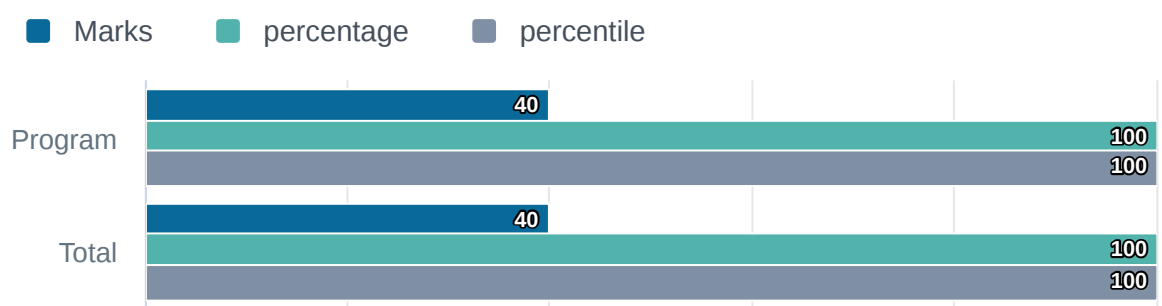
Ramya.V

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

Overall Summary

40	Marks Scored	out of 40
100 %	100 percentile	out of 41881 Test Takers
2m 44s	Time taken	of 1hr 20mins

Marks Scored



Attempt Summary

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

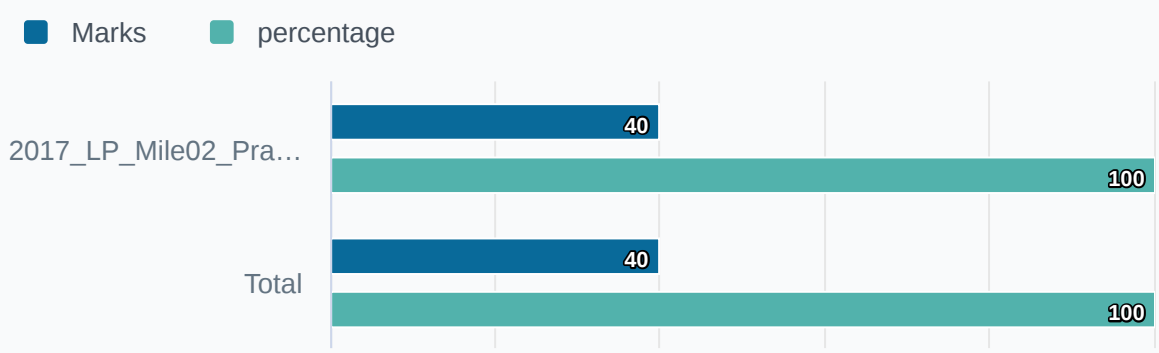
This shows the correctness of questions attempted by the test taker

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

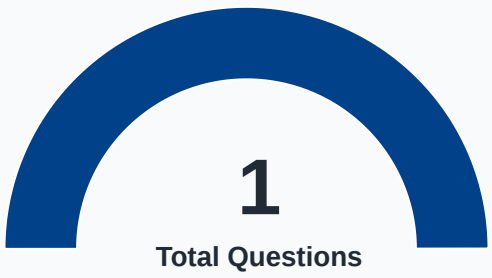
Section 1	question(s)	Time taken	Marks Scored
Program	1 Q.	2m 44s (Untimed)	40 / 40

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

07:23 PM	Started the test with Program
07:25 PM	Away from test window
07:25 PM	Away from test window
07:25 PM	Away from test window
07:25 PM	Away from test window
07:26 PM	Away from test window
07:26 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.

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Identify possible words: Detective Bakshi while solving a case stumbled upon a letter which had many words whose one character was missing i.e. one character in the word was replaced by an underscore. For e.g."Fi_er". He also found thin strips of paper which had a group of words separated by colons, for e.g. "Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer". He could figure out that the word whose one character was missing was one of the possible words from the thin strips of paper. Detective Bakshi has approached you (a computer programmer) asking for help in identifying the possible words for each incomplete word.

You are expected to write a function to identify the set of possible words.
The function **identifyPossibleWords** takes two strings as input
where,
input1 contains the incomplete word, and
input2 is the string containing a set of words separated by colons.
The function is expected to find all the possible words from **input2** that can replace the incomplete word **input1**, and return the result in the format suggested below.

Example1 -
input1 = "Fi_er"
input2 = "Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer"
output string should be returned as "FILER:FIXER:FIBER"

Note that –

- The output string should contain the set of all possible words that can replace the incomplete word in **input1**
- all words in the output string should be stored in UPPER-CASE

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 String identifyPossibleWords(String input1,String input2){
9          // Read only region end
10     {
11 String st[]=input2.split(":");
12         String str="";
13         for(int i=0;i<st.length;i++)
14         {
15             if(st[i].length()!=input1.length())
16                 continue;
17             String test=input1;
18             int x=test.indexOf("_");
19             char ch=st[i].charAt(x);
20             test=test.replace('_', ch);
21             test=test.toUpperCase();
22             st[i]=st[i].toUpperCase();
23             if(st[i].equals(test))
24             {
25                 if(str=="")
26                 {
```

Use Custom Input

Compile and Test | Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Identify possible words: Detective Bakshi while solving a case stumbled upon a letter which had many words whose one character was missing i.e. one character in the word was replaced by an underscore. For e.g."Fi_er". He also found thin strips of paper which had a group of words separated by colons, for e.g.
"Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer". He could figure out that the word whose one character was missing was one of the possible words from the thin strips of paper. Detective Bakshi has approached you (a computer programmer) asking for help in identifying the possible words for each incomplete word.

You are expected to write a function to identify the set of possible words.
The function **identifyPossibleWords** takes two strings as input
where,
input1 contains the incomplete word, and
input2 is the string containing a set of words separated by colons.
The function is expected to find all the possible words from **input2** that can replace the incomplete word **input1**, and return the result in the format suggested below.

Example1 -
input1 = "Fi_er"
input2 = "Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer"
output string should be returned as "FILER:FIXER:FIBER"

Note that –

- The output string should contain the set of all possible words that can replace the incomplete word in **input1**
- all words in the output string should be stored in UPPER-CASE

JAVA7 | Compiler: Java - 1.7

```
16                  continue;
17              String test=input1;
18              int x=test.indexOf("_");
19              char ch=st[i].charAt(x);
20              test=test.replace('_', ch);
21              test=test.toUpperCase();
22              st[i]=st[i].toUpperCase();
23              if(st[i].equals(test))
24              {
25                  if(str=="")
26                  {
27                      str=str+test;
28                  }
29                  else
30                  {
31                      str=str+":"+test;
32                  }
33              }
34          }
35          if(str=="")
36              str="ERROR-009";
37          return str;
38          }
39      }
40  }
41
```

Use Custom Input

Compile and Test | Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Identify possible words: Detective Bakshi while solving a case stumbled upon a letter which had many words whose one character was missing i.e. one character in the word was replaced by an underscore. For e.g."Fi_er". He also found thin strips of paper which had a group of words separated by colons, for e.g.
"Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer". He could figure out that the word whose one character was missing was one of the possible words from the thin strips of paper. Detective Bakshi has approached you (a computer programmer) asking for help in identifying the possible words for each incomplete word.

You are expected to write a function to identify the set of possible words.
The function **identifyPossibleWords** takes two strings as input
where,
input1 contains the incomplete word, and
input2 is the string containing a set of words separated by colons.
The function is expected to find all the possible words from **input2** that can replace the incomplete word **input1**, and return the result in the format suggested below.

Example1 -
input1 = "Fi_er"
input2 = "Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer"
output string should be returned as "FILER:FIXER:FIBER"

Note that –

- The output string should contain the set of all possible words that can replace the incomplete word in **input1**
- all words in the output string should be stored in UPPER-CASE

default

CODE EXECUTION DETAILS

Time: 226 ms

Memory: 103812 kb

TEST CASE INFORMATION

Input

Fi_er,Fever:filer:Filter:Fixer:fiber:fibre:tailor:offer

Expected Output

FILER:FIXER:FIBER

Actual Output

FILER:FIXER:FIBER

CONSOLE OUTPUT

STANDARD ERROR/WARNING

None

default