LP_Practice_AdditionUsingStrings

Ramya.V | 12 Feb 2023

Test Name::

LP_Practice_AdditionUsingStrings

Finish State: Browsing tolerance exceeded | Test Taken on: February 12, 2023 01:12:30 PM IST

R

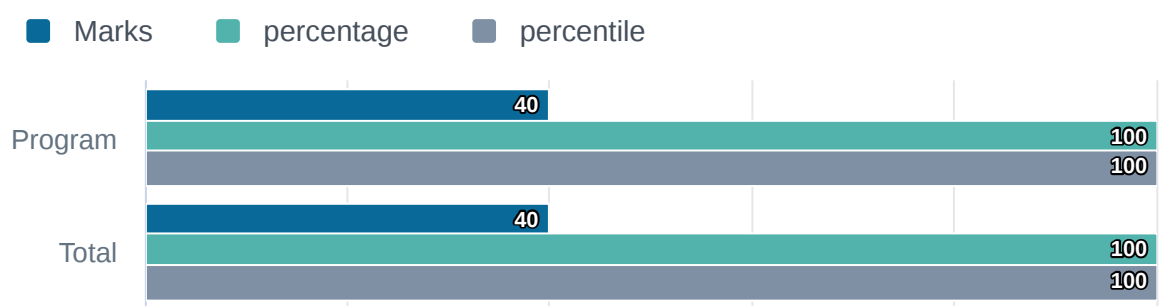
Ramya.V

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

Overall Summary

40	Marks Scored	out of 40
100 %	100 percentile	out of 45422 Test Takers
4m	Time taken	of 1hr 20mins

Marks Scored



Attempt Summary

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

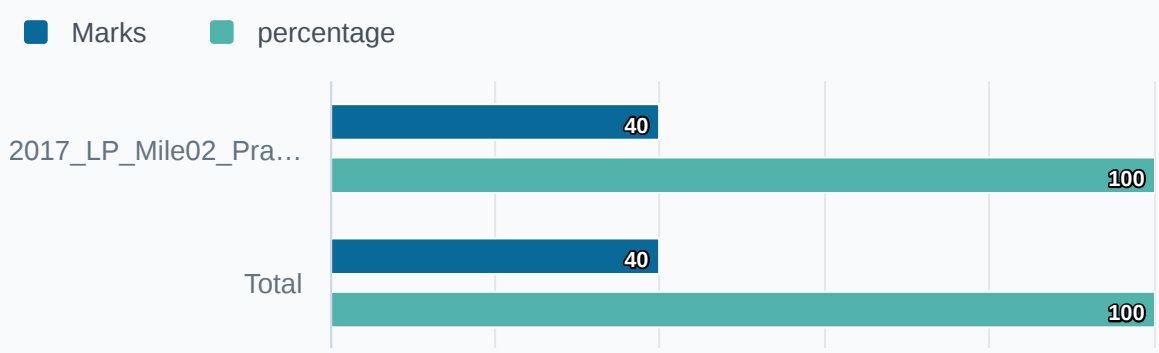
This shows the correctness of questions attempted by the test taker

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

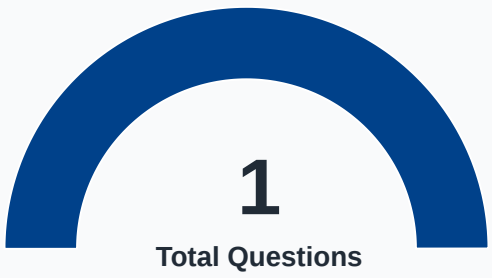
Section 1	question(s)	Time taken	Marks Scored
Program	1 Q.	4m (Untimed)	40 / 40

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

01:05 PM	Started the test with Program
01:05 PM	Away from test window
01:07 PM	Away from test window
01:08 PM	Away from test window
01:08 PM	Away from test window
01:09 PM	Away from test window
01:09 PM	Test ended (crossed browsing tolerance)

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?

Addition using Strings: Write a function that takes two numbers in string format and forms a string containing the sum (addition) of these two numbers.

Assumption(s):

- The input strings will contain only numeric digits
- The input strings can be of any large lengths
- The lengths of the two input string need not be the same
- The input strings will represent only positive numbers

For example –

- If input strings are "1234" and "56", the output string should be "1290"
- If input strings are "56" and "1234", the output string should be "1290"
- If input strings are "12345673212898954321 9" and "987612673489652", the output string should be "123457719741663032871"

NOTE: In Java & C#, this logic can be easily implemented using BigInteger

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 addNumberStrings(String input1,String input2){
9          // Read only region end
10     {
11 int carry=0;
12   if(input1.length()<input2.length())
13   {
14     String temp="";
15    temp=input1;
16    input1=input2;
17    input2=temp;
```

Use Custom Input

Compile and Test

Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Addition using Strings: Write a function that takes two numbers in string format and forms a string containing the sum (addition) of these two numbers.

Assumption(s):

- The input strings will contain only numeric digits
- The input strings can be of any large lengths
- The lengths of the two input string need not be the same
- The input strings will represent only positive numbers

For example –

- If input strings are "1234" and "56", the output string should be "1290"
- If input strings are "56" and "1234", the output string should be "1290"
- If input strings are "12345673212898954321 9" and "987612673489652", the output string should be "12345771974166303287 1"

NOTE: In Java & C#, this logic can be easily implemented using BigInteger

JAVA7 | Compiler: Java - 1.7

```
18    }
19    int len1=input1.length();
20    int len2=input2.length();
21    String str="";
22    int j=len2-1;
23    for(int i=0;i<len1;i++)
24    {
25     int a=Character.getNumericValue(input1.charAt(len1-1-i));
26     int b=0;
27     if(j>=0)
28     {
29      b=Character.getNumericValue(input2.charAt(j));
30      j--;
31     }
32     int sum=a+b+carry;
33     carry=sum/10;
34     int init=sum%10;
```

Use Custom Input

Compile and Test | Submit Code

1. Program

1

Attempted: 1/1

Question 1

Revisit Later

How to Attempt?

Addition using Strings: Write a function that takes two numbers in string format and forms a string containing the sum (addition) of these two numbers.

Assumption(s):

- The input strings will contain only numeric digits
- The input strings can be of any large lengths
- The lengths of the two input string need not be the same
- The input strings will represent only positive numbers

For example –

- If input strings are "1234" and "56", the output string should be "1290"
- If input strings are "56" and "1234", the output string should be "1290"
- If input strings are "12345673212898954321 9" and "987612673489652", the output string should be "123457719741663032871"

NOTE: In Java & C#, this logic can be easily implemented using BigInteger

JAVA7

Compiler: Java - 1.7

```
30      j--;
31     }
32     int sum=a+b+carry;
33     carry=sum/10;
34     int init=sum%10;
35     str=Integer.toString(init)+str;
36     if(i==len1-1 && carry>0)
37     {
38             str=Integer.toString(carry)+str;
39      }
40    }
41    return str;
42    }
43   }
44  }
45
46
```

Use Custom Input

Compile and Test

Submit Code