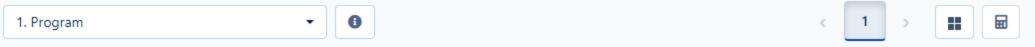
YUKESH R M

(i)



## Question 1

☐ Revisit Later

## How to Attempt?

digitSum: The labels on a trader's boxes display a large number (integer). The trader wants to label the boxes with a single digit ranging from 1 to 9. He decides to perform digit sum on this large number, continuously till he gets a single digit number.

NOTE: In mathematics, the "digit sum" of a given integer is the sum of all its digits, (e.g.: the digit sum of 84001 is calculated as 8+4+0+0+1 = 13, the digit sum of 13 is 1+3 = 4).

Write a function (method) that takes as input a large number and returns a single digit by performing continuous digitSum on this number, and on the resulting numbers, till the resulting number is a single digit number in the range 1 to 9.

**Example 1:** If the large number whose single-digit digitSum is to be found is 976592, the process is as below – 9+7+6+5+9+2=38

3+8 = 11

1+1=2

Thus, the single-digit digitSum for the number 976592 is 2.

**Example 2:** If the large number whose single-digit digitSum is to be found is 123456, the process is as below – 1+2+3+4+5+6 = 21

2+1=3

Thus, the single-digit digitSum for the number 123456 is 3.

For negative numbers, the result should also be in negative.

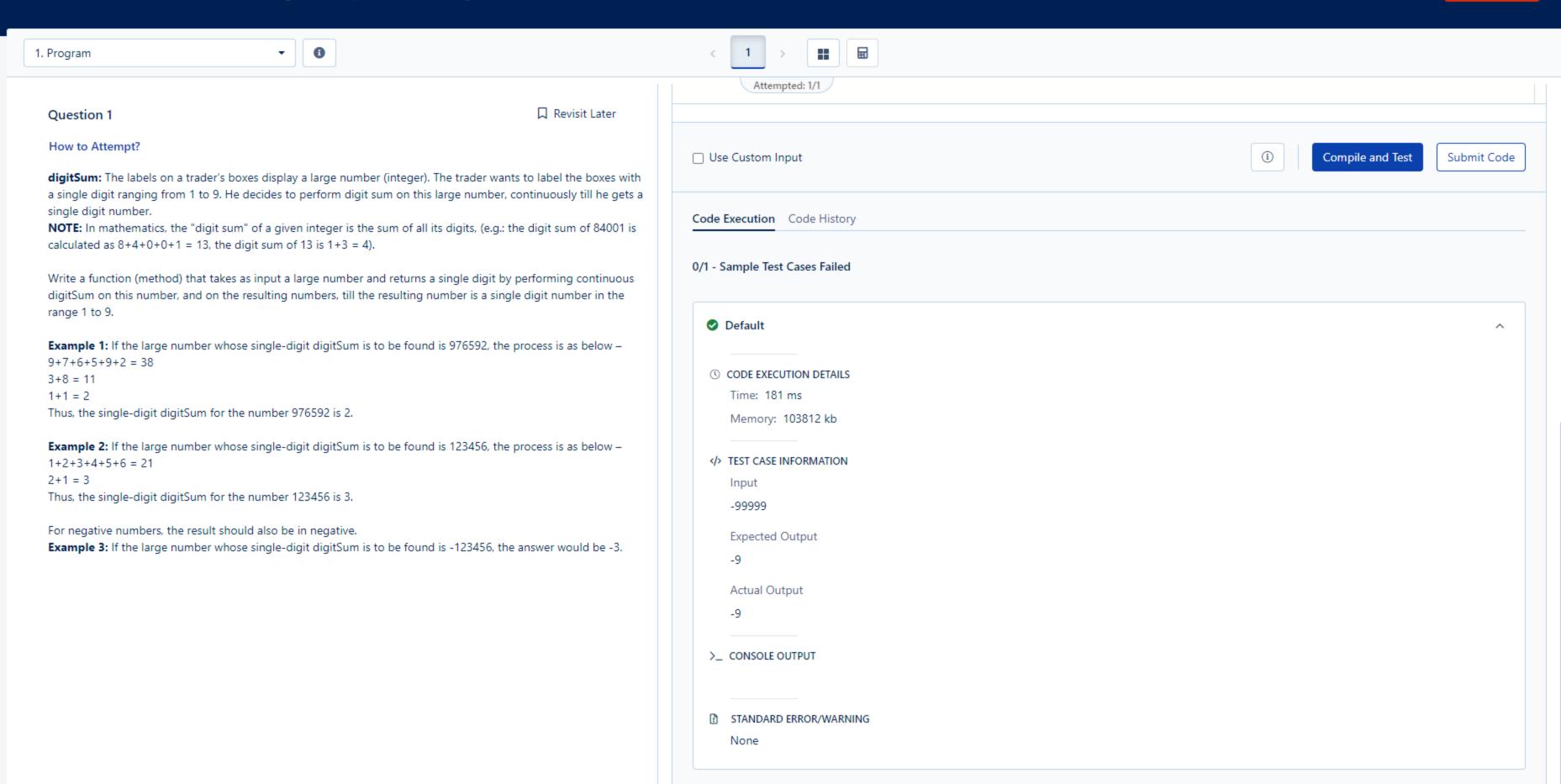
**Example 3:** If the large number whose single-digit digitSum is to be found is -123456, the answer would be -3.

```
Attempted: 1/1
                                                                                                              5 ( 4>
JAVA7
                               Compiler: Java - 1.7
       import java.io.*;
       import java.util.*;
       // Read only region start
        class UserMainCode
   6
           public int digitSum(int input1){
               // Read only region end
  10
               int digitCount = String.valueOf(input1).length();
               if (digitCount == 1) return input1;
  11
  12
  13
               int sum = 0;
  14
               int number = input1;
  15
               if (input1 < 0) number *= -1;</pre>
  16
  17
               while (digitCount != 1) {
  18
                   int currentSum = 0;
  19
                    while (number != 0) {
                        currentSum += number % 10;
  20
  21
                        number /= 10;
  22
  23
                   sum = currentSum;
  24
                   number = sum;
  25
                    digitCount = String.valueOf(sum).length();
  26
  27
  28
               if (input1 < 0) sum *= -1;
  29
               return sum;
  30
  31
```

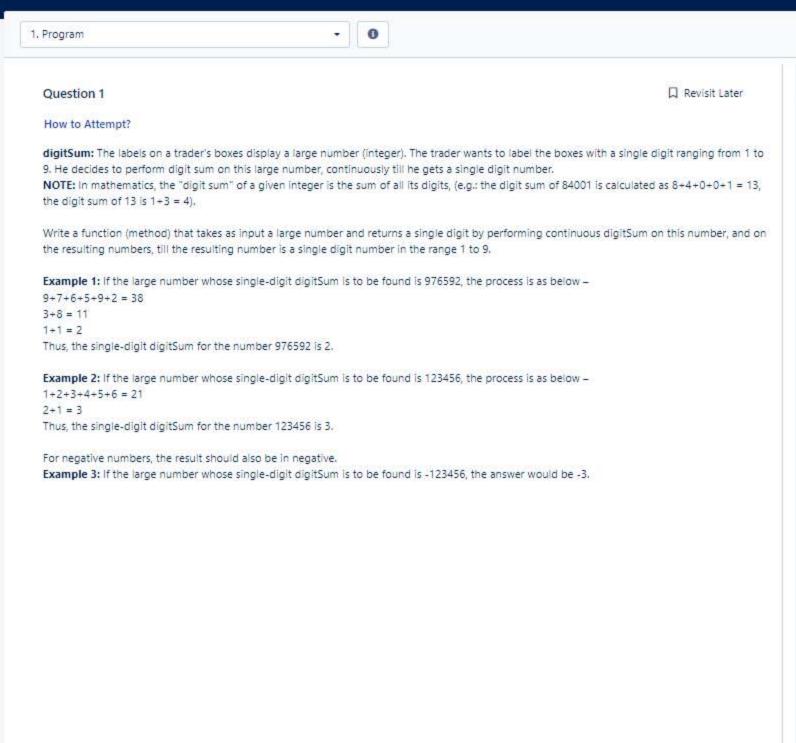
☐ Use Custom Input

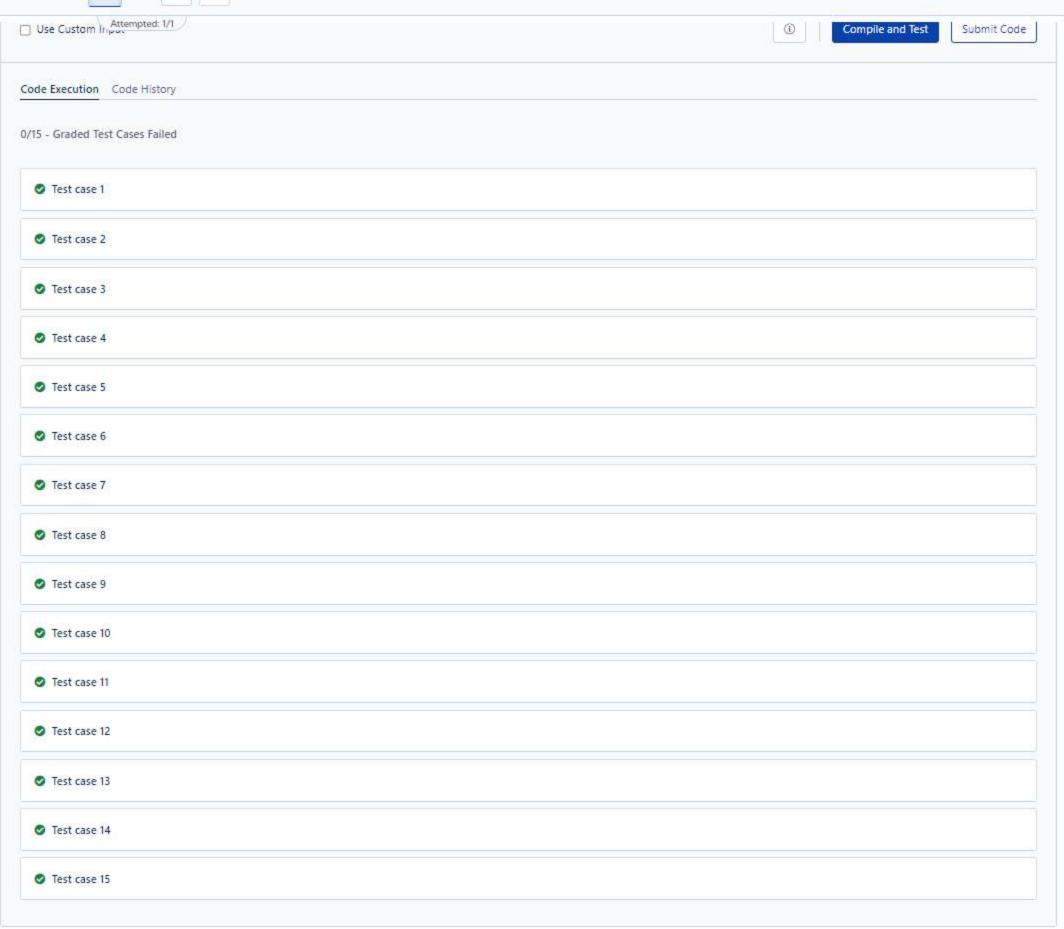
Compile and Test

Submit Code

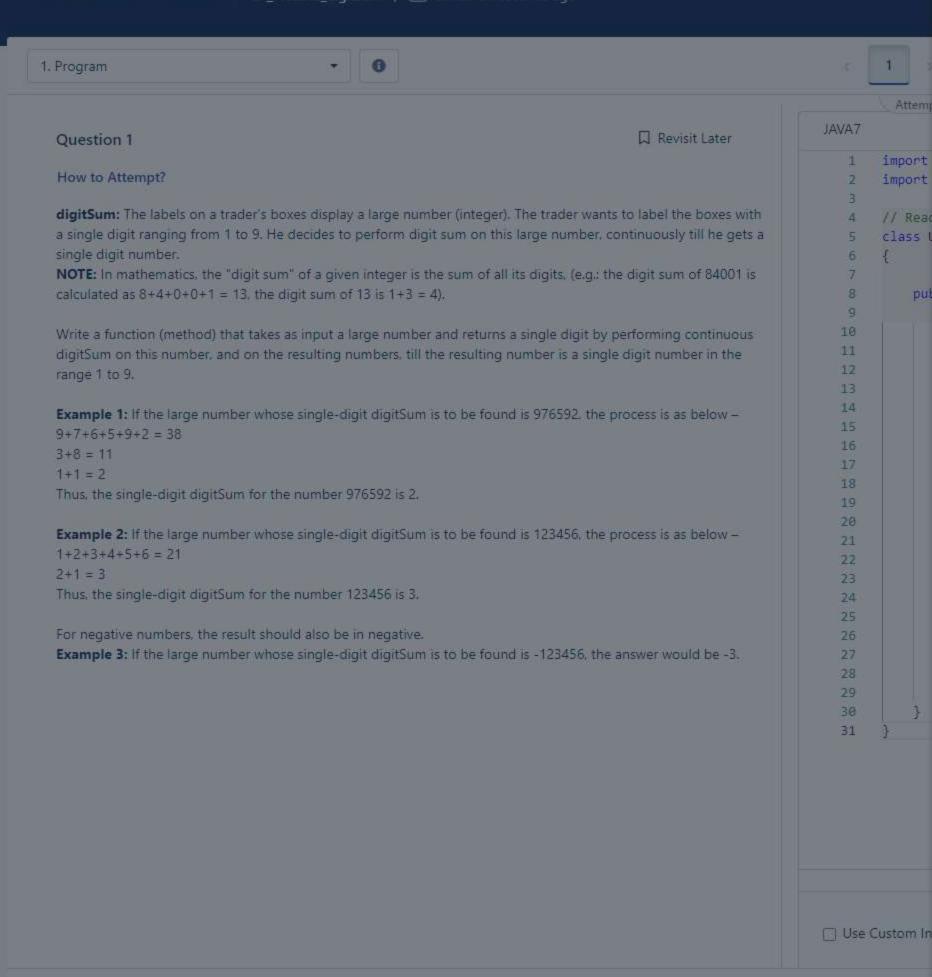


**== +91 80471-90902** 





T +91 80471-90902





## Section Summary

A Finish Test

#	SECTION NAME	STATUS	
1.	Program Untimed Section	1	0
			Total: 1 Questions