Recursion: Sum of Two Numbers (with a twist!)

Published by Deep Xavier in Java

algorithms math recursion strings

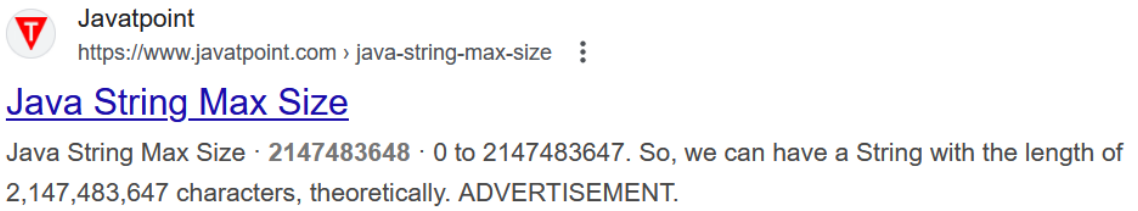
This is an "**expert**" challenge??!! Why is a sum of two numbers an "**expert**" challenge??!! Well, the numbers can have 1000 digits or even beyond such count...

So, what's the **twist**? You have to do the summation as if you're doing it manually on a piece of paper, thus, the conversion of the **numeric string** to **numeric literal** is basically **disallowed**.

Examples

Once again, I have been drawn into a recursive challenge.
I have chosen to undertake this since I hit an anomaly with my previous palindromic example.
And once again, this challenge has several same traits. It is expecting that the variables provided are maintained in their native format.
However this challenge will require a subtle conversion of String to integer

LOGIC
The problem is correct in stating that there are no boundaries:



I will start with a more simplified arrangement

```
sum("4666", "544") → "5210"
```

Assume String number1 = 4666
String number2 = 544

lengthNumber1 = number1.length = 4;
lengthNumber2 = number2.length = 3;

We know substring is zero index based
We also know that it will exclude the last value passed into it.
For instance substring(0,6) would be index 0 to 5
It is best to use single index in substring since it will take all values from this index onwards.

Single digit is of interest when performing addition:
number1.substring(index here should be start position only, which is index of the last digit)
number1.substring(number1.length) = index 4 = array out of bound exception)
number1.substring((number1.length-1)) = index 3 = **6**

Same principle for number2.substring((number2.length-1) = index 2 = **4**

So on each recursive this revised substring is passed into the method.

<u>Questions arising so far:</u>
* It will calculate the addition and store digit in a String total.
* This String will be passed across in recursive call.
* There is no need to pass the original number across since the total System.out.println() can occur in non-recursive method such as main.
* When the length of either substring exceeds the other, it will perform its last addition as per usual. It would either carry the remainder across as per usual. Otherwise if there is no remainder, it would simply drop the outstanding digits from the longer number downwards.
* So far I do not envisage any issues at all, but I know I have more than likely not foreseen most issues that this problem will entail.