

# Big Sum

---

Anik, a student of mathematics is doing a research for his PhD degree. He thinks if he can complete his work, it will help mankind a lot. But the problem is as part of his research he needs to check sum of two numbers in a special way. The numbers themselves will be big enough plus the sum needs to be done in a reversed way. For example, if two numbers were 24 and 13, we have to first reverse these numbers to make them 42 and 31 and then we have to do the addition so the sum will be  $42 + 31 = 73$ . Afterwards he needs to output the result as reversed again, so the output will be 37 actually. One advantage he has is that the input numbers will never have leading or trailing zero. But the resulted sum if has any leading zeroes need to be removed. For example, if we have two numbers such as 911 and 199, then the reversed numbers will be 119 and 991 and the sum will be  $119 + 991 = 1110$ . So, if we reverse, the output should be 0111, but actually we have to consider 111 as output because we should remove all leading zeroes in final output. As it is hard for Anik to check numbers that can be as long as 100 digits, so he wants to hire a freelancer who can write a small program to help him. He will provide a large input file with these number pairs and expects an output file that finds out the reversed sum. He wants the code as well so that he can run it on other data sets in future. Let's see whether you can do it or not.

## Input specification:

Input will consist of many number pairs, each on a separate line and the numbers are separated by a comma. The numbers can be as big as 100 digits long. The numbers will be integer and factional numbers and negative numbers will not be in input. Input will be terminated by end of file.

## Output specification:

The output will be the reversed sums each on an individual line of its own. For each input pair, there should be one reversed sum.

## Sample Input:

```
199,911
119,991
24,13
```

## Sample Output:

```
111
111
37
```