Submissions

Testrun
0pt | Not attempted **0/4 users** correct (0%)

encoded_sum
6pt | Not attempted **13/13 users** correct (100%)
11pt | Not attempted **12/12 users** correct (100%)

air_show
5pt | Not attempted **14/14 users** correct (100%)
20pt | Not attempted **1/4 users** correct (25%)

toothpick_sculpture
10pt | Not attempted **9/10 users** correct (90%)
15pt | Not attempted **0/3 users** correct (0%)

gold
15pt | Not attempted **6/10 users** correct (60%)
18pt | Not attempted **4/6 users** correct (67%)

Top Scores

bmerry	65
sevenkplus	65
fhlasek	65
mnbvmar	65
eatmore	52
Merkurev	47
ikatanic	37
tozangezan	32
tmt514	32
wafrelka	22

Problem B. encoded_sum

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 6 points *2 minute timeout*	The contest is finished.
large 11 points *10 minute timeout*	The contest is finished.

Problem

Encoded Sum

You are an archaeologist studying a lost civilization. This civilization used a decimal (base 10) number system like our own: their numbers were made up of digits from 0 through 9, with the most significant digit on the left. However, this civilization used the ten letters A through J, in some order, to represent the ten digits 0 through 9, in a one-to-one mapping.

You have just discovered two scrolls from that civilization. Each contains a number representing the population of one of the two regions of the civilization. The numbers are of the same length. You do not know the civilization's letter-to-digit mapping, but you know it is consistent across the two documents. You want to know the maximal possible sum of those two numbers. Please note that the resulting numbers can have leading zeros.

Given these two scrolls, return the maximal possible sum. Since the output can be a really big number, we only ask you to output the remainder of dividing the result by the prime 10^9+7 (1000000007).

Input

The input library is called "encoded_sum"; see the sample inputs below for examples in your language. It defines three methods:

- **GetLength()**:
  - Takes no argument.
  - Returns a 64-bit integer: the length of the number on either scroll.
  - Expect each call to take 0.1 microseconds.
- **GetScrollOne(i)**:
  - Takes a 64-bit number in the range $0 \le i <$ GetLength()
  - Returns an uppercase character (in the inclusive range A through J) representing the i-th character in the first scroll, counting starting from 0, starting from the left.
  - Expect each call to take 0.1 microseconds.
- **GetScrollTwo(i)**:
  - Takes a 64-bit number in the range $0 \le i <$ GetLength()
  - Returns an uppercase character (in the inclusive range A through J) representing the i-th character in the second scroll, counting starting from 0, starting from the left.
  - Expect each call to take 0.1 microseconds.

Output

Output a single line with a single integer: the maximal sum modulo the prime 10^9+7 (1000000007), as described above.

Limits

Number of nodes: 100. **(Notice that the number of nodes is the same for both the Small and Large datasets.)**
Time limit: 3 seconds.
Memory limit per node: 128 MB.
Maximum number of messages a single node can send: 1000.
Maximum total size of messages a single node can send: 8 MB.
$1 \le$ GetLength() $\le 10^9$.

Small dataset

GetScrollOne(i) is either uppercase A or uppercase B, for all i.
GetScrollTwo(i) is either uppercase A or uppercase B, for all i.

Large dataset

GetScrollOne(i) is an uppercase letter between A and J, inclusive, for all i.
GetScrollTwo(i) is an uppercase letter between A and J, inclusive, for all i.

Sample

Input	Output
See input files below.	For sample input 1: 17 For sample input 2: 1888 For sample input 3: 395283726

Sample input libraries:
Sample input for test 1: encoded_sum.h [CPP] encoded_sum.java [Java]
Sample input for test 2: encoded_sum.h [CPP] encoded_sum.java [Java]
Sample input for test 3: encoded_sum.h [CPP] encoded_sum.java [Java]

All problem statements, input data and contest analyses are licensed under the Creative Commons Attribution License.

© 2008-2017 Google Google Home - Terms and Conditions - Privacy Policies and Principles

Powered by

Google Cloud Platform