

3. Сейф (seif)

1 секунда

100 очков

Археологи нашли на затонувшем корабле сейф, у которого очень своеобразный цифровой замок. На экране замка видны два целых числа одинаковой длины, которые время от времени меняются. Исследователи заметили, что изменения состояния на экране бывают трёх видов:

- к обоим числам справа добавляется по одной цифре;
- у обоих чисел справа стирается какое-то количество цифр;
- на экране появляется третье число и рядом с ним поле ввода размером в одну цифру.

У исследователей возникла гипотеза, что для открытия сейфа в поле ввода нужно ввести ту цифру суммы двух больших чисел, позицию которой показывает третье число. Так как данные на экране меняются слишком быстро для вычисления в уме или на бумаге, то нужна программа, которая решила бы эту проблему.

Основной компонент этой программы реагирует на события на экране сейфа. Твоя задача написать этот компонент, следующие функции которого вызываются в общем N раз:

Функция	Описание
<code>void Add(int f, int s)</code>	К первому длинному числу справа добавилась цифра f , а ко второму цифра s .
<code>void Del(int k)</code>	Из обоих длинных чисел пропало k цифр справа. Можно учитывать, что всегда в момент вызова этой функции большие числа будут по крайней мере k -значными. Может случиться, что в результате этой функции в них не останется цифр.
<code>int Get(int p)</code>	На экране появилось третье число p . Функция должна вернуть из суммы двух больших чисел p -тую цифру справа. Можно учитывать, что всегда на момент вызова этой функции сумма будет по крайней мере p -значной.

В тестовой среде есть файлы-примеры, где нужные функции уже описаны и как решение нужно записать только их реализации. Также в файл решения можно записывать и свои функции. Для тестирования своего решения на своём компьютере есть также и пример оценивающей программы, описание входных и выходных данных которой приведено ниже (на сервере используется другая оценивающая программа, которая проверяет также и верность ответов, возвращаемых решением). Для компиляции и тестирования своего решения на своём компьютере:

Язык	Решение	Команды
C++	<code>seif.cpp</code>	<code>g++ -o main main.cpp seif.cpp ./main</code>
Java	<code>seif.java</code>	<code>javac -cp . main.java seif.java java -cp . main</code>
Python	<code>seif.py</code>	<code>python main.py</code>

Входные данные. Первая строка текстового файла `seifsis.txt` содержит количество вызовов функций компонента N ($1 \leq N \leq 10^6$), а каждая из следующих N строк — описание одного вызова:

- Строка вида $1 \ F \ S$ ($1 \leq F \leq 9, 1 \leq S \leq 9$) означает вызов `Add(f, s)`.
- Строка вида $2 \ K$ означает вызов `Del(k)`.
- Строка вида $3 \ P$ означает вызов `Get(p)`.

Выходные данные. В текстовый файл `seifval.txt` выводится лог общения оценивающей программы и компонента.

Пример.	<code>seifsis.txt</code>	<code>seifval.txt</code>
	5	add(1, 1)
	1 1 1	get(1) = 2
	3 1	add(3, 9)
	1 3 9	get(1) = 2
	3 1	get(2) = 3
	3 2	

После первой операции на экране видны числа 1 и 1, а их сумма конечно 2. После третьей операции на экране видны числа 13 и 19, а их сумма 32.

Пример.	<code>seifsis.txt</code>	<code>seifval.txt</code>
	10	add(1, 1)
	1 1 1	add(2, 2)
	1 2 2	add(3, 3)
	1 3 3	get(1) = 6
	3 1	get(2) = 4
	3 2	get(3) = 2
	3 3	del(2)
	2 2	add(4, 6)
	1 4 6	get(1) = 0
	3 1	get(2) = 3
	3 2	

После третьей операции на экране видны числа 123 и 123, а их сумма 246. После седьмой операции на экране видны числа 1 и 1, а после восьмой числа 14 и 16, сумма которых 30.

Оценивание. В тестах общей стоимостью 25 дополнительно выполняется условие $N \leq 1000$. В остальных тестах никаких дополнительных ограничений нет.