Задача 7. Найти коллизию

Источник:	основная
Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	3 секунды
Ограничение по памяти:	разумное

В данной задаче вам предлагается найти коллизию для неизвестной вам хеш-функции, то есть указать два различных ключа, на которых значение хеш-функции совпадает.

Известно, что хеш-функция принимает 32-битное беззнаковое целое число на вход (ключ) и выдаёт 32-битное беззнаковое целое число на выход (хеш). Кроме того, известно, что хеш-функция очень хорошего качества.

Вы можете вычислять хеш-функцию на любых ключах, на каких хотите. Однако всего разрешается сделать не более $2 \cdot 10^5$ вычислений.

Протокол взаимодействия

В данной задаче ваша программа будет работать не с файлами, а совместно с программой-интерактором. Ваша программа и интерактор будут запускаться одновременно, и соединяться пайпами (теми самыми пайпами, о которых упоминалось на лекции при рассмотрении очереди и кольцевого буфера). Всё, что ваша программа выводит в `stdout`, читает интерактор из своего `stdin`, а всё, что пишет интерактор на `stdout`, читает ваша программа из своего `stdin`.

Ваша программа должна печатать команды, которые интерактор будет выполнять. Есть два типа команд:

- Команда `eval`, после которой через пробел должно быть записано 32-битное беззнаковое целое число X. Эта команда предписывает интерактору вычислить значение хеш-функции от числа X. Интерактор вычислят его и записывает искомый хеш: ваша программа должна прочитать его из `stdin` как беззнаковое 32-битное целое.

- Команда `answer`, после которой через пробел должно быть записано два 32-битных беззнаковых целых числа A и B. Этой командой ваша программа должна сообщить интерактору коллизию: числа A и B должны быть различными, но их хеш должен совпадать. После этого ваша программа должна сразу же завершить исполнение, ничего никуда больше не записывая и ничего ниоткуда не читая.

Если вы сделаете больше вычислений хеш-функции, чем разрешено, или выдадите неверный ответ командой `answer`, то ваше решение получит Wrong Answer.

Поскольку задача интерактивная, требуется:

1. Не открывать никаких файлов, **не** использовать `freopen` и `fopen`.
2. Писать команды с помощью `printf` и читать ответы с помощью `scanf`.
3. После каждой команды выводить символ перевода строки и сразу после этого выполнить: `fflush(stdout);`

Если вы забудете сделать команду `fflush`, то записанные вами в `stdout` данные останутся в буфере, и никогда не попадут в пайп, а значит интерактор никогда их не получит и всё зависнет (вердикт Timeout).

Учтите, что в этой задаче все числа беззнаковые, так что писать и читать их надо с форматом `"%u"`.

Пример

stdin	stdout
2478003845 894250524 622810134 894250524	eval 1 eval 2 eval 3 eval 4 answer 2 4

Пояснение к примеру

Обратите внимание, что в примере сначала программа печатает команды в `stdout`, а уже потом на них приходят ответы от интерактора. В данном случае у ключей 2 и 4 получается одинаковый хеш, равный 894250524.

Исполняемый файл интерактора вы можете скачать по ссылке (только для Windows). Чтобы запустить его просто поиграться, нужно использовать командную строку:

```
interactor.exe input.txt output.txt
```

Чтобы запустить его вместе с вашим решением `sol.exe`, можно использовать командную строку (предварительно надо поставить Python 3):

```
python run_interactive.py sol.exe
```

Выведенные вашим решением команды будут записаны в `output.txt`.