

A. Укоротитель URL

	Все языки	Oracle Java 8
Ограничение времени	3 секунды	3 секунды
Ограничение памяти	128Mb	400Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

Необходимо реализовать укоротитель URL (URL Shortener), который преобразует адреса URL в более короткие.

Например, из оригинального длинного URL `https://stackoverflow.com/questions/4527942/comparing-two-dictionaries-and-checking-how-many-key-value-pairs-are-equal` делает `https://prakti.kum/56yf`, где:

- `https://prakti.kum/` — наш веб-сервис укоротителя URL, этот префикс всегда остаётся неизменным.
- `56yf` — строка, зашифрованный алгоритмом Base62 порядковый номер URL в базе данных укоротителя.

На вход программе могут приходить запросы двух видов.

- `post оригинальный_URL` : необходимо сократить URL, вернуть укороченный, и сохранить соответствие в базу данных. Если данный URL уже есть в базе данных, то надо вернуть имеющееся соответствие, а не обновлять укороченный URL.
- `get укороченный_URL` : необходимо вернуть оригинальный URL, если соответствующая запись имеется в базе данных. Иначе вернуть "error". (обратите внимание, что, вообще говоря, в `get`-запрос могут передать любой URL, даже не из нашего укоротителя)

Описание функции шифрования алгоритма Base62, преобразующей число `num` (порядковый номер URL) в короткую шифр-строку.

- Заведём константы — строку `BASE="0123456789ABCDEFGHIJKLMNQRSTUWXYZabcdefghijklmnopqrstuvwxyz"` и число `B=62`
- Присвоим `q = num, encoded_string = ""`
- Пока `q > 0`:
 - Считаем остаток от деления `q` на `B`, `r = q mod B`
 - Считаем целую часть от деления, `q = int(q / B)`
 - Берём символ по индексу `r` и дописываем его к строке `encoded_string = BASE[r] + encoded_string`
- Возвращаем `encoded_string`

Пример работы функции шифрования алгоритма Base62

Преобразуем число `num=46777`:

1. `q = 46777, r = 29, encoded_string = "T"`
2. `q = 754, r = 10, encoded_string = "AT"`
3. `q = 12, r = 12, encoded_string = "CAT"`
4. `q = 0`

Ответ: "CAT"

Формат ввода

В первой строке записано целое число $k \leq 10^9$, начиная с которого будут отсчитываться номера добавляемых URL (см. пример).

Во второй строке записано количество запросов $n \leq 10^5$.

В следующих n строках записаны запросы в одном из двух форматов — «post URL» либо «get URL». URL — строка не длиннее 2048 символов. URL может состоять из маленьких и больших латинских букв, цифр, скобок, символов «=:?#&\$_-./».

Формат вывода

Выведите ответ на каждый запрос в отдельной строке.

Пример

Ввод

```
1000007
4
post https://yandex.com/images/programming/image_id_sbcjsdbjb7486584ehjvdbf
post https://yandex.com/images/cats/image_id_sbcjsdbjb7486584ehjddf7f7
get https://prakti.kum/4C99
get https://prakti.kum/4C100
```

Вывод

```
https://prakti.kum/4C99
https://prakti.kum/4C9A
https://yandex.com/images/programming/image_id_sbcjsdbj
error
```

Примечания

Пояснения к примеру:

1. Пришёл запрос `post`, нужно вернуть укороченный URL. Порядковый номер нового URL = 1000007. Зашифруем это число, получим "4C99". Запишем полученную строку после доменного имени нашего веб-сервиса — `https://prakti.kum/4C99`
2. Пришёл запрос `post`. Добавим следующий URL под номером 1000008, получим `https://prakti.kum/4C9A`
3. Пришёл запрос `get`, нужно вернуть оригинальный URL. Проверим, выдавали ли мы ранее короткий URL `https://prakti.kum/4C99`. Вернём соответствующий оригинальный URL.
4. Пришёл запрос `get`, нужно вернуть оригинальный URL. Проверим, выдавали ли мы ранее короткий URL `https://prakti.kum/4C100`. Нет — поэтому вернём сообщение об ошибке "error".