

# Ловушка поезда

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

*Это интерактивная задача.*

Однажды, в один из своих обычных дней (не «обычных», полных путешествий по временным линиям, а обычных), Дэдпул решил отправиться на расслабляющую поездку в поезде — да-да, тот самый поезд, который пересекает всю страну. Не раздумывая, он встал на платформе и сел в первый попавшийся вагон.

Уэйд наслаждался видами из окна. Поездка шла, как ни странно, гладко. Но через несколько часов Дэдпул начал замечать нечто странное: он был уверен, что видел этот же красный амбар за окном не так давно. Немного подумав, он понял, что на самом деле это не обычный день, а «обычный» и УВИ заперли его в циклическом поезде (то есть, в поезде, где можно вернуться к тому же вагону, из которого начался путь, двигаясь в одном направлении). Пытаясь перейти в соседний вагон, он осознал ещё одну проблему — абсолютно все вагоны выглядели **одинаково**, за исключением одной кнопки, которая переключала состояние обогревателя в вагоне. Изначально Дэдпулу ничего не было известно о состояниях обогревателей в вагонах.

Дэдпул понял, что для того, чтобы выбраться из этой ловушки, ему нужно определить количество вагонов в поезде. У него есть следующие действия:

1. «F» — перейти в следующий вагон;
2. «D» — перейти в предыдущий вагон;
3. «C» — потрогать обогреватель в текущем вагоне (чтобы определить его состояние — включен или выключен);
4. «S» — нажать кнопку и поменять состояние обогревателя (то есть, выключить обогреватель, если он включен, и включить, если он выключен).

Помогите Дэдпулу определить количество вагонов в поезде. Поскольку он хочет сделать это как можно быстрее, вы должны потратить **не более**  $16 \cdot n$  действий, где  $n$  — количество вагонов в поезде.

## Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка входных данных содержит одно целое число  $t$  — количество наборов входных данных ( $1 \leq t \leq 3000$ ).

Гарантируется, что сумма ответов (количества вагонов) по всем наборам входных данных не превосходит 3000.

## Протокол взаимодействия

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Вы можете выполнить одно из действий, описанных в условии, не более  $16 \cdot n$  раз. Для некоторых действий вы получите ответы:

1. «F» — перейти в следующий вагон. Ответа от интерактора не последует;
2. «D» — перейти в предыдущий вагон. Ответа от интерактора не последует;
3. «C» — потрогать обогреватель в текущем вагоне. Интерактор выведет 0, если обогреватель выключен, и 1 в противном случае.
4. «S» — нажать кнопку. Ответа от интерактора не последует.

Чтобы вывести ответ на задачу, напечатайте «!  $n$ », где вместо  $n$  должно быть предложенное вами количество вагонов в поезде ( $1 \leq n \leq 3000$ ). Этот вывод не учитывается в количестве запросов. После этого интерактор выведет вердикт — 1, если ваше предположение верно, и 0 в противном случае (в этом случае ваша программа получит вердикт **WA** — Wrong Answer).

Если в какой-то момент ваша программа превышает лимит в  $16 \cdot n$  запросов, ваша программа завершится с вердиктом **WA** (Wrong Answer).

**Важно:** не забывайте после каждого запроса сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт **TL** (Time Limit) или **IL** (Idleness Limit).

**Пример**

стандартный ввод	стандартный вывод
1	C
0	F C
1	F C
0	S C
1	D C
1	D C
1	! 2
1	