

Добрый эксперимент

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Известный алхимик Шу Такер специализируется на создании химер. Известно, что «рецепты» (алгоритмы создания) химер кодируются правильными скобочными последовательностями.

Напомним, что скобочная последовательность из скобок '(' и ')' называется *правильной*, если ее можно получить из корректного арифметического выражения удалением всего, кроме скобок. Более формально определим рецепт химеры, то есть правильную скобочную последовательность, следующим образом:

- пустая строка является базовым рецептом;
- если t_1 и t_2 — два рецепта химер, то новую химеру можно получить их объединением; ее рецептом будет конкатенация рецептов $t_1 t_2$ (второй рецепт, выписанный сразу за первым);
- если t — рецепт химеры, то другую химеру можно получить алхимическим преобразованием с рецептом (t) ;
- любая другая последовательность, которую нельзя получить с помощью описанных правил, не является рецептом химеры.

Так, например, $\langle (((()))) \rangle$ получается алхимией из $\langle () (()) \rangle$, которая, в свою очередь, получается объединением из химер $\langle () \rangle$ и $\langle (()) \rangle$. Обе из них получаются из базового пустого рецепта алхимией.

Рецепт последнего достижения Шу Такера, разговаривающей химеры, можно представить как правильную скобочную последовательность s длины n . Это несомненно потрясающее достижение, и рецепт строго засекречен, однако у Эдварда и Альфонса есть определенные сомнения на счет использованных материалов...

Эдвард может делать запросы формата «? l r », и на каждый такой запрос узнавать, является ли отрезок рецепта $s_l s_{l+1} \dots s_r$ правильной скобочной последовательностью. Иными словами, если выписать только символы s с l -го по r -й включительно, будет ли полученная скобочная последовательность удовлетворять данному выше определению.

Не превышая разрешенное число запросов, полностью раскройте секретный рецепт s .

Формат входных данных

В первой строке ввода даны два целых числа n и q — длина последовательности и максимальное число разрешенных запросов, соответственно ($1 \leq n, q \leq 2 \cdot 10^5$; n четно; $q \geq n - 3$).

После ввода этих значений запускается протокол взаимодействия с интерактором.

Формат выходных данных

В конце взаимодействия с интерактором выведите ваше предположение о том, чему равна строка s , в формате, описанном в протоколе взаимодействия с интерактором.

Протокол взаимодействия

Гарантируется, что к моменту запуска протокола взаимодействия с интерактором строка s зафиксирована, и что q запросов достаточно.

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора. Вы можете выполнить запрос, описанный в условии, не более q раз, где q зависит от подзадачи.

Чтобы сделать запрос, выведите строку «? l r » ($1 \leq l \leq r \leq n$) в отдельной строке. После чего также на отдельной строке интерактор выведет «YES», если соответствующий отрезок s является правильной скобочной последовательностью, и «NO» иначе.

Чтобы вывести ответ на задачу, выведите строку «! t », где t — ваше предположение о том, чему равна загаданная s . После этого протокол взаимодействия с интерактором завершается. Во избежание получения некорректных вердиктов программа вашего решения тоже должна завершиться. Вывод ответа на задачу не учитывается в счет общего числа сделанных запросов.

Если в какой-то момент ваша программа превышает лимит в q запросов, или значения l или r в очередном запросе выходят за границы s , интерактор завершится с вердиктом WA (Wrong Answer).

Важно: не забывайте после каждой выведенной строки 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 Exceeded) или IL (Idleness Limit Exceeded).

Система оценки

Баллы за подзадачи начисляются только в случае, если все тесты соответствующей подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
0	—	примеры из условия		полная
1	5	$n = 6; q = 100$	0	первая ошибка
2	6	$n = 10; q = 10\,000$	0, 1	первая ошибка
3	7	$n = 22; q = 59\,000$	0 – 2	первая ошибка
4	10	«глубина» s не превышает 2; $q = 2 \cdot 10^5$		первая ошибка
5	10	$n = 500; q = 2 \cdot 10^5$	0 – 3	первая ошибка
6	16	$n = 10\,000; q = 2 \cdot 10^5$	0 – 3, 5	первая ошибка
7	8	$n = 500; q = 500$	0 – 3, 5	первая ошибка
8	12	$q = n$	0 – 7	первая ошибка
9	6	$q = n - 1$	0 – 8	первая ошибка
10	7	$q = n - 2$	0 – 9	первая ошибка
11	13	$q = n - 3$	0 – 10	первая ошибка

Под *глубиной* последовательности в четвертой группе подразумевается максимальная разность между числом открывающих и закрывающих скобок на любом отрезке от начала s до произвольной другой позиции.

Примеры

стандартный ввод	стандартный вывод
6 100 YES YES	? 1 4 ? 3 6 ! () ()
6 100 NO NO YES	? 1 4 ? 3 6 ? 3 4 ! (())