

## Задача А. Помогите, спасите!

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

### Формат входных данных

В единственной строке входных данных содержится непустая строка  $S$ , состоящая из  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ) маленьких букв английского алфавита.

### Формат выходных данных

Выведите  $N$  строк, в  $i$ -й строке должно содержаться количество различных подстрок в  $i$ -м префиксе строки  $S$ .

### Примеры

стандартный ввод	стандартный вывод
aabab	1 2 5 8 11
atari	1 3 5 9 14

## Задача В. Общие подстроки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано  $K$  не обязательно различных строк из маленьких латинских букв, с суммарной длиной  $N$ .  $L_i$  определяется как максимальная длина строки, которая встречается как подстрока хотя бы у  $i$  строк из начального набора. Требуется для каждого  $2 \leq i \leq K$  посчитать  $L_i$ .

### Формат входных данных

В первой строке входных данных дано одно число  $L$  ( $1 \leq L \leq 200\,000$ ) — число строк.

В следующих  $L$  строках даны сами строки из начального набора, по одной в строке. Гарантируется, что  $N$  — суммарная длина всех строк не превышает 200 000.

### Формат выходных данных

В  $k - 1$  строке выведите по одному числу —  $L_2, L_3, \dots, L_K$ .

### Пример

стандартный ввод	стандартный вывод
6	5
matter	3
animate	2
pattern	2
thermal	1
domain	
teammate	

## Задача С. Ненокку

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

### Формат входных данных

В каждой строчке входного файла записана одна из двух записей.

1. ? <слово> (<слово> — это набор не более 50 латинских символов): запрос проверки существования подстроки <слово> в произведении;
2. A <текст> (<текст> — это набор не более  $10^5$  латинских символов): добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более  $10^5$  символов. Суммарная длина всех запросов не превосходит 15 мегабайт плюс 12140 байт.

### Формат выходных данных

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

### Пример

стандартный ввод	стандартный вывод
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

## Задача D. Циклический сдвиг

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Дана строка  $s$ , состоящая из маленьких латинских букв.

Назовем строку  $t = t_1t_2 \dots t_m$  ( $m > 0$ ) хорошей относительно строки  $s$ , если строка  $t$  и ее левый циклический сдвиг  $t' = t_2 \dots t_mt_1$  являются подстроками строки  $s$ .

Вам необходимо найти количество различных хороших строк  $t$  относительно заданной строки  $s$ .

### Формат входных данных

В единственной строке входных данных записана строка  $s$ , состоящая из  $n$  ( $1 \leq n \leq 300\,000$ ) маленьких латинских букв.

### Формат выходных данных

Выведите единственное число — количество хороших строк  $t$  относительно заданной строки  $s$ .

### Примеры

стандартный ввод	стандартный вывод
abaac	7
aaa	3

### Замечание

В первом примере хорошими строками являются следующие: a, b, c, aa, ab, ba, aba.

Во втором примере хорошими являются только три строки: a, aa, aaa.

## Задача Е. Словарь: перезагрузка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 1024 мегабайта

*Словарь* — это множество слов. Вы должны уметь обрабатывать запросы трех типов:

- «+ word» — добавить слово **word** в словарь, если оно в нем не присутствует.
- «- word» — удалить слово **word** из словаря, если оно там присутствует.
- «? text» — вычислить суммарное количество вхождений всех слов из словаря в текст **text**, при этом, если слово входит в текст несколько раз, то необходимо учесть каждое вхождение.

Гарантируется, что любое слово или текст являются непустыми строками, состоящими из букв **a**, **b** и **c**, суммарная длина которых не превосходит  $L$ . Однако, для упрощения задачи перед выполнением каждого запроса необходимо поступить следующим образом: пусть  $x$  обозначает ответ на последний запрос **?**, или 0, если таких запросов еще не было. Тогда необходимо очередную строку (**word** или **text**) циклически сдвинуть  $x$  раз. Напомним, что циклическим сдвигом строки  $s = s_0s_1 \dots s_{|s|}$  называется строка  $s' = s_1 \dots s_{|s|}s_0$ .

### Формат входных данных

В первой строке дано одно число  $Q$  — число запросов. В следующих  $Q$  строках находятся запросы. Суммарная длина строк во всех запросах не превосходит  $L$  ( $L \leq 5\,000\,000$ )

### Формат выходных данных

Для каждого запроса «?» выведите одно число — ответ на него.

### Пример

стандартный ввод	стандартный вывод
11	0
+ a	6
+ a	5
- a	7
- ab	
? abca	
+ ab	
+ a	
? abaaabb	
? aaabbab	
+ baa	
? babaca	

## Задача F. Рефрен HARD

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Рассмотрим последовательность  $n$  целых чисел от 1 до  $m$ . Подпоследовательность подряд идущих чисел называется рефреном, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 750\,000$ ,  $1 \leq m \leq 10$ ).

Вторая строка содержит  $n$  целых чисел от 1 до  $m$ .

### Формат выходных данных

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

### Пример

стандартный ввод	стандартный вывод
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

## Задача G. Задача для разминки рук

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 6.5 секунд  
Ограничение по памяти: 768 мегабайт

Дано дерево на  $n$  вершинах с корнем в вершине 1. В  $i$ -й вершине записан символ  $t_i$  - одна из трёх латинских букв a, b, c. Также дана строка  $s$  длины  $m$ , состоящая из строчных латинских букв a, b, c.

Каждой паре вершин  $(u, v)$  естественным образом можно сопоставить строку, которая получается последовательным выписыванием символов в вершинах на единственном простом пути от  $u$  до  $v$ , начиная с символа в вершине  $u$ .

Вам нужно посчитать количество пар целых чисел  $(u, v)$  таких, что  $1 \leq u, v \leq n$ , и строка, соответствующая пути от  $u$  до  $v$ , лексикографически меньше или равна  $s$ .

Деревом называется связный граф, где у каждой вершины кроме первой (первая вершина называется корнем) задан единственный предок, с которым вершина соединена ребром. Путём в дереве между вершинами  $u$  и  $v$  называется такая последовательность вершин, в которой  $u$  — первая вершина,  $v$  — последняя, и любые 2 подряд идущие вершины соединены ребром. Путь называется простым, если никакая из вершин не встречается в нем дважды.

Строка  $a$  считается лексикографически меньше строки  $b$ , если существует такое число  $k$ , что на всех позициях меньших  $k$  строки  $a$  и  $b$  совпадают, а на  $k$ -й позиции символ строки  $a$  лексикографически меньше соответствующего символа строки  $b$ , или же в том случае, если длина строки  $a$  меньше длины строки  $b$ , а все символы на одинаковых позициях у строк совпадают.

### Формат входных данных

В первой строке вводятся два целых числа  $n$  и  $m$  ( $2 \leq n \leq 10^6$ ,  $1 \leq m \leq 10^6$ ) — число вершин в дереве и длина строки для сравнения.

Во второй строке без пробелов вводятся  $m$  символов  $s_1s_2 \dots s_m$  — строка для сравнения.

В третьей строке без пробелов вводятся  $n$  символов  $t_1t_2 \dots t_n$ ,  $i$ -й из которых обозначает букву, записанную в вершине  $i$ .

В следующей строке вводится  $(n - 1)$  число  $p_2, p_3, p_4 \dots, p_n$  ( $1 \leq p_i \leq n$ ), где  $p_i$  обозначает номер предка вершины  $i$  в дереве.

Гарантируется, что каждый символ в данных строках  $s$  и  $t$  равен одной из строчных латинских букв «a», «b», «c», и что заданный граф действительно образует дерево.

### Формат выходных данных

Выведите единственное число — искомое количество пар.

### Примеры

стандартный ввод	стандартный вывод
3 2 ba acb 3 1	5
5 3 bac abcab 1 1 3 1	18
2 3 acc bb 1	0

## Замечание

В первом примере всего есть 9 пар чисел от 1 до 3. Парам (1, 1), (1, 2), (1, 3), (3, 3), (3, 1) соответствуют строки **a**, **abc**, **ab**, **b**, **ba**. Парам, где первое число 2, соответствуют строки, начинающиеся на **c** — эти строки в любом случае лексикографически больше строки **ba**. Паре (3, 2) соответствует строка **bc**, которая лексикографически больше **ba**.

Во втором примере всего есть 25 пар. Первым числом в паре не может быть 3, так как в вершине 3 записана буква **c**, которая меньше первой буквы **s**. Также не подходят пары (2, 4) и (5, 4).

В третьем примере все символы в вершинах больше первой буквы **s**, поэтому ни одна пара вершин не подходит.

## Система оценки

Тесты к этой задаче состоят из 12 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$m$		
0	0	–	–	–	Тесты из условия.
1	11	$n \leq 500$	$m \leq 500$	0	
2	8	$n \leq 5000$	$m \leq 5000$	0, 1	
3	13	$n \leq 30\,000$	$m \leq 500$	0, 1	Нет путей длины больше 500.
4	10	$n \leq 100\,000$	$m \leq 100\,000$	–	$p_i = i - 1$
5	14	$n \leq 100\,000$	$m \leq 100\,000$	–	$s$ и $t$ состоят из символов « <b>a</b> ».
6	8	$n \leq 100\,000$	$m \leq 100\,000$	0–5	
7	5	$n \leq 200\,000$	$m \leq 200\,000$	0–6	
8	5	$n \leq 350\,000$	$m \leq 350\,000$	0–7	
9	6	$n \leq 500\,000$	$m \leq 500\,000$	0–8	
10	6	$n \leq 650\,000$	$m \leq 650\,000$	0–9	<b>Offline-проверка.</b>
11	7	$n \leq 800\,000$	$m \leq 800\,000$	0–10	<b>Offline-проверка.</b>
12	7	–	–	0–11	<b>Offline-проверка.</b>



## Задача Н. Суффиксный пулемёт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Теоретическая подготовка новобранцев армии Поссилтума включала в себя не только занятия по военному праву, но и начала криптографии. Лекции читал майор Мега Байт, не чуждый солдатского юмора. Гвидо и Нунцио, в чьё задание входил развал армии Поссилтума изнутри, решили на этом сыграть, внося путаницу в терминологию. В начале очередной лекции Нунцио поднял руку и спросил:

— Вот вы на прошлой лекции рассказывали про конечные автоматы. А про конечные пулемёты расскажете?

Мега Байт не растерялся.

— Суффиксный пулемёт — это конечный автомат, принимающий все суффиксы данной строки (от нулевого до  $L$ -го включительно, где  $L$  — длина строки), и только их. Сержант Гвидо!

— Я, господин майор!

— Вы сможете отличить автомат от пулемёта?

— Так точно, господин майор!

— Вам дан конечный автомат. Требуется проверить, является ли он суффиксным пулемётом данной строки.

К сожалению, написание программ такого типа не входило в обязанности Гвидо и Нунцио как в Синдикате, так и в корпорации М. И. Ф. Так что соответствующую программу придётся писать Вам.

### Формат входных данных

Во входном файле задан один или несколько тестовых наборов. В первой строке каждого набора заданы количество состояний автомата  $N$ , количество переходов  $M$ , а также количество принимающих состояний  $T$  ( $1 \leq T \leq N \leq 50\,000$ ,  $1 \leq M \leq 100\,000$ ). Во второй строке через пробел заданы  $T$  различных чисел в пределах от 1 до  $N$  — принимающие состояния автомата, в возрастающем порядке. В последующих  $M$  строках заданы переходы в виде  $a_i b_i c_i$ , где  $1 \leq a_i, b_i \leq n$ , а  $c_i$  — маленькая буква латинского алфавита. Переход производится из состояния  $a_i$  в состояние  $b_i$  по букве  $c_i$ . Из каждого состояния  $a_i$  есть не более одного перехода по символу  $c_i$ . Последняя строка описания набора — это строка  $S$ , для которой автомат должен являться пулемётом. Она состоит только из маленьких латинских букв, и её длина лежит в пределах от 1 до 50 000 включительно. Кроме того, сумма всех  $N$  и суммарная длина всех строк, для которых необходимо произвести проверку, не превосходит 50 000, а сумма всех  $M$  не превосходит 100 000.

Файл заканчивается фиктивным набором, в котором  $N = M = T = 0$ .

Начальным состоянием автомата является первое. Если при интерпретации какой-то строки в автомате отсутствует соответствующий переход, то автомат вываливается по ошибке и строку не принимает. Таким образом, строка принимается, только если при её интерпретации были найдены все переходы, и по их завершении автомат оказался в принимающем состоянии (при этом неважно, были по пути принимающие состояния, или нет).

### Формат выходных данных

Выведите в выходной файл, является ли данный автомат пулемётом, следуя формату примера.

## Пример

стандартный ввод	стандартный вывод
2 1 2	Automaton 1 is a machinegun.
1 2	Automaton 2 is not a machinegun.
1 2 a	
a	
2 2 2	
1 2	
1 1 a	
1 2 b	
ab	
0 0 0	

## Задача I. Вставить текст

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Алиса и Аня работают копирайтерами. Недавно им пришёл заказ: нужно написать  $k$  текстов (строк) на схожую тематику. Девочки сразу приступили к работе и быстро получили  $k$  строк  $S_1, \dots, S_k$ , каждая из которых имеет длину не более  $m$  и состоит из строчных латинских букв, при чем длина  $S_1$  оказался в точности равна  $m$ .

Девочки уже давно работают вместе, и у них есть простой способ проверить оригинальность своей работы. Оригинальность оценивается с помощью подстрок полученных текстов. Алиса и Аня считают некоторую строку неоригинальной, если и она, и строка, получающаяся из данной разворотом, встречаются в некоторых из их текстов в качестве подстрок на одних и тех же позициях.

Чтобы ускорить проверку, они разбивают строки на блоки. Разбиение на  $t$  блоков задается последовательностью  $a_0, a_1, \dots, a_t$ , где  $a_0 = 0$ ,  $a_t = m$  и  $a_{i-1} < a_i$  для любого  $1 \leq i \leq t$ . Тогда  $i$ -м блоком называется отрезок целых чисел  $[a_{i-1} + 1; a_i]$ . Блок  $[a_{i-1} + 1; a_i]$  называется интересным, если существует неоригинальная строка, встречающаяся в текстах ровно на позициях, задаваемых данным блоком. Иными словами, блок — интересный, если для каких-то строк  $S_l$  и  $S_r$  (возможно,  $l = r$ ) верно, что  $|S_l|, |S_r| \geq a_i$ , и строка  $S_{l, a_{i-1}+1} S_{l, a_{i-1}+2} \dots S_{l, a_i}$  совпадает со строкой  $S_{r, a_i} S_{r, a_i-1} \dots S_{r, a_{i-1}+1}$ , где  $S_{t,j}$  —  $j$ -й слева символ строки  $S_t$ .

Например, для текстов [abba, ba] последовательности  $(0, 1, 4)$  и  $(0, 1, 2, 3, 4)$  задают корректные разбиения, а последовательности  $(1, 2, 3)$  и  $(0, 1, 1, 4)$  — нет. При этом для разбиения  $(0, 2, 4)$  первый блок  $[1; 2]$  является интересным, поскольку  $S_{1,1} S_{1,2} = S_{2,2} S_{2,1} = ab$ , а второй блок  $[3; 4]$  — нет, поскольку для единственной возможной пары номеров  $l = r = 1$  строки  $S_{1,3} S_{1,4} = ba$  и  $S_{1,4} S_{1,3} = ab$  не совпадают.

Разбиение называется интересным, если каждый блок в этом разбиении интересный. Алиса и Аня хотят найти интересное разбиение текстов на минимальное число блоков, чтобы измерить оригинальность работы. Девочки постарались, чтобы условие этой задачи было современным и прошло тест Бекдел, поэтому теперь помогите им и напишите для них программу, измеряющую оригинальность!

Заметим, что искомая величина корректно определена, так как, разбив строки на  $m$  блоков длины 1, мы получим интересное разбиение (в каждом блоке будет достаточно взять  $l = r = 1$ ).

### Формат входных данных

Во первой строке вводятся три целых числа  $t$ ,  $k$  и  $m$  ( $1 \leq k \leq 200\,000$ ,  $1 \leq m \leq 500\,000$ ) — номер группы, к которой относится данный тест, общее число текстов и длина первого текста.

В  $i$ -й из следующих  $k$  строк вводится  $S_i$  —  $i$ -й текст, состоящий из строчных букв латинского алфавита.

Гарантируется, что  $|S_1| = m$ ,  $|S_i| \leq m$  для любого  $i > 1$ , и суммарная длина всех строк не превосходит 500 000.

### Формат выходных данных

Во первой строке выведите единственное целое число  $t$  — минимальное число блоков в интересном разбиении.

Во второй строке через пробел выведите возрастающую последовательность из  $t - 1$  целого числа  $a_1, \dots, a_{t-1}$  — номеров правых границ всех блоков, кроме последнего (само разбиение имеет вид  $[1; a_1], [a_1 + 1; a_2], \dots, [a_{t-1} + 1; m]$ ).

## Примеры

стандартный ввод	стандартный вывод
0 2 6 abcded cba	2 3
0 6 7 poggers sus amogus tokyo ghoul sodluv	4 3 5 6

## Замечание

В первом примере вторая строка `cba` при перевороте совпадает с первыми тремя символами первой строки `abcded`. Оставшиеся же символы `ded` образуют палиндром, т.е. эта строка совпадает с собой же перевёрнутой. Поэтому мы можем разбить строки на два блока  $[1; 3]$  и  $[4; 6]$ . Легко видеть, что на меньшее число блоков разбить нельзя, ведь `abcded` — не палиндром.

Во втором примере в первом блоке  $[1; 3]$  можно выбрать строку-палиндром `sus`, во втором блоке кусочки пятой (`ghoul`) и шестой (`sodluv`) строк, совпадающих друг с другом при перевороте. В третьем и четвертом блоках выбираем по одной букве из любой строки. Можно показать, что на меньшее число блоков разбить строки нельзя.

## Система оценки

Тесты к этой задаче состоят из десяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Обозначим суммарную длину всех строк как  $L$ .

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		$k$	$m$	$L$		
0	0	–	–	–	–	Тесты из условия.
1	7	$k \leq 100$	$m \leq 10$	$L \leq 1000$	0	
2	8	$k \leq 20$	$m \leq 100$	$L \leq 2000$	0	
3	10	$k \leq 200$	$m \leq 200$	$L \leq 40\,000$	0 – 2	
4	12	$k \leq 200$	$m \leq 2000$	$L \leq 400\,000$	0 – 3	
5	9	–	–	–	–	Гарантируется, что в ответе не больше двух блоков.
6	11	$k = 1$	$m \leq 50\,000$	$L \leq 50\,000$	–	
7	11	$k = 1$	–	–	6	
8	10	$k = 2$	–	–	–	
9	9	–	$m \leq 200\,000$	$L \leq 200\,000$	0 – 3, 6	
10	13	–	–	–	0 – 9	<b>Offline-проверка.</b>

## Задача J. Суффиксный путь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В одной супер-секретной лаборатории «Кимод йтысяед» было разработано супер-секретное устройство. Синдикату «Черное солнце» удалось выяснить, что устройство имело супер-секретное название «Тамотва 173». Лучшие умы пытались расшифровать его, но так ничего и не удавалось. Пробовали и шифр цезаря, и сдвиг, и RSA. В конце концов, синдикат обратился за помощью в Весенний Гуманитарный Детский сад (ВГД). Как оказалось, это — также супер-секретная организация, которая занимается подготовкой элитного отряда дворников. Они профессионально умеют складывать мусор на самых видных местах, подметать так, что просыпается весь район и также немного увлекаются философией. Разумеется, они мгновенно поняли, что в названии зашифровано ни что иное, как основная часть этого устройства! Еще они выяснили, что это устройство умеет обрабатывать только файлы размера не более, чем шестьдесят мегабайт.

После этого синдикат обратился за помощью к лучшим друзьям ВГД, Ежедневному Женевскому Завтраку (ЕЖЗ). Они также специализируются на шифрах: красят заборы, моют полы, а также занимаются модернизацией и инновациями. ЕЖЗ сообщили, что это устройство — их профиль, оно идеально сочетается с их идеологией. Более формально, устройство также умеет выполнять модернизацию. К сожалению, с инновациями может справиться только ЕЖЗ, поэтому устройство не умеет их придумывать, а может лишь проверять, является ли некоторая инновация действительно революционной инновацией. Как оказалось, внутри устройства хранится строка, состоящая из маленьких латинских букв. Модернизация состоит в том, чтобы к хранящейся строке дописать маленькую латинскую букву. ЕЖЗ хотели сообщить более подробную информацию, но после фразы, что ВГД ошиблись в подсчете максимального размера обрабатываемого файла в тридцать раз, они были в непригодном для общения состоянии: у большинства отвалилась челюсть.

Синдикат обратился к своим последним друзьям: команде Инноваций и Культа Лени (ИКЛ). Они сразу объяснили, что строка является для устройства инновационной, если она является частью хранящейся в устройстве строки. Но всем сразу стало ясно: здесь есть какой-то подвох! И снова лучшие умы стали биться над этой задачей. Перепробовали всё: и кричать, и танцевать, и петь песенки, и бить баклуши. Но так им и не удавалось понять, что же происходит на самом деле. Наконец, директор ИКЛ пришёл к директору ЕЖЗ, и они, вместе с директором ВГД и мокренькой кисонькой, поняли, что строка должна быть не просто частью, а, будучи развернутой, должна являться префиксом развернутой строки, хранящейся в устройстве! Более того, она также должна состоять из маленьких латинских букв. После этого все вместе они пошли спать.

Из более достоверных источников (а именно Мадагаскарский Национальный Отряд Профессиональных Супер-агентов - МНОП) стало известно, что тот, у кого окажется данное устройство, получит неограниченную власть над миром. Ваша задача кристально ясна: реализуйте данное устройство.

### Формат входных данных

В первой строке входного файла содержится число  $M$  ( $1 \leq M \leq 5 \cdot 10^5$ ) — количество выполняемых операций. Далее в  $M$  строках содержатся описания операций: либо  $+$  с для модернизации, либо  $?$  s для проверки инновационности. Гарантируется, что устройство сможет обработать входной файл.

### Формат выходных данных

Для каждого запроса проверки выведите «YES», если строка является инновационной, и «NO» в противном случае.

## Пример

стандартный ввод	стандартный вывод
2 + a ? a	YES