

Задача А. Ахо

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

Это интерактивная задача.

Каждый раз, когда Маргарет и Альфред посещают кафе «У Дональда», они играют в странную игру про угадывание строк.

Её правила таковы:

- Первый игрок пишет строку S фиксированной длины N . Также у первого игрока есть строка T , изначально пустая. Обе строки состоят только из маленьких букв английского алфавита.
- Второй игрок не знает эти строки в течение всей игры. Однако ему разрешено спрашивать про любые две позиции (в обеих строках), правда ли, что символы в них равны. Например, вопрос может выглядеть так “Равны ли второй символ строки S и пятый символ строки T ?” Обратите внимание, что можно спрашивать про два символа одной строки.
- Игра состоит из M раундов. В начале каждого раунда первый игрок добавляет один символ в конец строки T .
- После добавления символа второй игрок может задать не более пяти вопросов. После этого он должен сказать, какое число подстрок строки T равно строки S .

Маргарет быстро заметила, что Альфред всегда преуспевает в роли второго игрока. Она подозревает наличие стратегии, позволяющей второму игроку выигрывать независимо от S и T . А вы так сможете?

Формат входных данных

При запуске ваша программа должна считывать два целых числа N и M ($1 \leq N, M \leq 20\,000$) из стандартного потока ввода.

Далее следуют M раундов игры. В i -м раунде вы можете задать **не более пяти** вопросов в формате “<позиция1> <позиция2>”. Описание любой позиции выглядит как “s x ” где $1 \leq x \leq N$ (если это x -й символ строки S) или как “t y ” где $1 \leq y \leq i$ (если это y -й символ строки T). Ответ программы жюри будет “Yes”, если символы на этих позициях равны и “No” иначе.

Формат выходных данных

В конце любого раунда вы должны вывести ответ в формате “\$ k ”, где k равно числу вхождений строки S в строку T . После этого в строку T будет автоматически добавлен новый символ (если это не последний раунд).

Не забудьте делать `flush` после каждого вопроса. После того, как вы вывели все m чисел, ваша программа должна автоматически завершиться, иначе ваш вердикт может быть каким угодно.

Примеры

стандартный ввод	стандартный вывод
3 7	s 1 s 2
No	s 1 s 3
Yes	s 2 t 1
No	s 1 t 1
Yes	\$ 0
Yes	s 2 t 2
Yes	\$ 0
No	s 3 t 3
No	\$ 1
Yes	s 2 t 4
Yes	s 1 t 4
Yes	\$ 1
	s 1 t 5
	\$ 1
	s 2 t 6
	\$ 1
	s 3 t 7
	\$ 2

Замечание

В примере, строка *S* изначально равна “aba”, а строка *T* получается добавлением символов “a”, “b”, “a”, “c”, “a”, “b”, “a”.

Задача В. Палиндромы

Имя входного файла: `palindrome.in`
Имя выходного файла: `palindrome.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, `abba` — палиндром, а `омах` — нет. Для строки α будем обозначать $\alpha[i..j]$ ее подстроку длины $j - i + 1$ с i -й по j -ю позицию включительно (позиции нумеруются с единицу). Для заданной строки α длины N ($1 \leq N \leq 100\,000$) требуется подсчитать число q пар (i, j) , $1 \leq i < j \leq n$, таких что $\alpha[i..j]$ является палиндромом.

Формат входных данных

Входной файл содержит одну строку α длины N , состоящую из маленьких латинских букв.

Формат выходных данных

В выходной файл выведите искомое число q .

Примеры

<code>palindrome.in</code>	<code>palindrome.out</code>
<code>aaa</code>	3
<code>abba</code>	2
<code>омах</code>	0

Задача С. Цензура

Имя входного файла: `censored.in`
Имя выходного файла: `censored.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Посчитайте, сколько строк над алфавитом из n символов длины m не содержат ни одной подстроки из заданного множества “запрещенных” строк.

Формат входных данных

В первой строке написаны целые числа n ($1 \leq n \leq 100$) — количество символов в алфавите, m ($1 \leq m \leq 100$) — длина искомых строк и p ($0 \leq p \leq 10$) — количество “запрещенных” подстрок. Следующая строка содержит n символов с кодами больше 32 — буквы алфавита. Далее идет p “запрещенных” строк, длины которых не превосходят $\min(m, 10)$ символов. Строки целиком состоят из символов алфавита.

Формат выходных данных

В первой строке выведите ответ на задачу.

Примеры

<code>censored.in</code>	<code>censored.out</code>
2 3 1 ab bb	5

Задача D. Башни

Имя входного файла: `towers.in`
Имя выходного файла: `towers.out`
Ограничение по времени: 10 секунды
Ограничение по памяти: 256 мегабайт

Задано число n и последовательность из n чисел. Требуется рассмотреть все возможные циклические сдвиги заданной последовательности, отсортировать их в лексикографическом порядке, и вывести сумму наибольших общих префиксов соседних в этом порядке сдвигов.

Формат входных данных

Входной файл содержит не более 200 тестовых примеров. Каждый тестовый пример состоит из двух строк. Первая из них содержит целое число $1 \leq n \leq 50000$ — количество магических башен. Вторая строка содержит n чисел в интервале от 0 до 100 — заданную последовательность.

После последнего тестового примера вместо числа n идет 0.

Формат выходных данных

Для каждого тестового примера выведите одно число — искомую сумму.

Примеры

<code>towers.in</code>	<code>towers.out</code>
11 12 8 18 18 8 18 18 8 15 15 8 0	13

Задача Е. Динамический поиск подстрок

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

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.

Примеры

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

Задача F. Формула в КНФ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	512 мегабайт

Ежегодный Парад за Равенство $P = NP$ уже совсем близко. Байтиана – лидер движения $P = NP$ movement, решила раз и навсегда показать своим противникам и опубликовать доказательство известного равенства.

Она докажет, что $P = NP$, предъявив полиномиальный алгоритм для известной NP -полной задачи под названием CNF-SAT. В этой задаче нам дано n булевых переменных x_1, \dots, x_n и булева формула в конъюнктивной нормальной форме. Эта формула имеет вида

$$(l_{1,1} \vee \dots \vee l_{1,q_1}) \wedge (l_{2,1} \vee \dots \vee l_{2,q_2}) \wedge \dots \wedge (l_{m,1} \vee \dots \vee l_{m,q_m}),$$

где каждое выражение вида $(l_{i,1} \vee \dots \vee l_{i,q_i})$ называется клаузой и каждое выражение $l_{i,j}$ называется литералом, например, одна из переменных x_1, \dots, x_n или ее отрицание. Будем считать, что корректная клауза не содержит два одинаковых литерала. Например, $(x_1 \vee \neg x_3) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_1 \vee x_2)$ является булевой формулой в конъюнктивной нормальной форме с $n = m = 3$.

Задача CNF-SAT интересуется, существует ли назначение переменных x_1, \dots, x_n , что заданная формула верна.

К сожалению, доказательству Байтианы не хватает одного шага. Она заявляет, что она смогла свести¹ общую задачу CNF-SAT к специальному случаю, где каждая клауза C является последовательной, то есть, обладает следующими свойствами:

- Для всех i , x_i и $\neg x_i$ не могут одновременно находиться в C .
- Если для i, j, k верно $i < j < k$ и C содержит переменную x_i (либо ее отрицание) и x_k (либо $\neg x_k$), то она также содержит x_j (или $\neg x_j$).

Например, если $n = 3$, клаузы (x_2) и $(\neg x_3 \vee \neg x_1 \vee x_2)$ являются последовательными; а $(x_2 \vee \neg x_2)$ и $(x_1 \vee \neg x_3)$ не являются.

Помогите Байтиане найти эффективный алгоритм решения этого специального случая CNF-SAT. Чтобы удивить ее еще сильнее, напишите программу, которая считает количество различных назначений переменных x_1, \dots, x_n , для которых заданная формула верна.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 1\,000\,000$) – число переменных. Вторая строка содержит описание формулы CNF-SAT, используя переменные x_1, \dots, x_n (однако, некоторые из них могут не присутствовать в формуле), состоящей только из последовательных клауз. Формула задана в следующем формате (посмотрите пример для лучшего понимания).

- Каждая клауза начинается с открывающей скобки (и заканчивается закрывающей скобкой).
- Литерал x_i (for $1 \leq i \leq n$) записывается как x_i , а литерал $\neg x_i$ записывается, как $\sim x_i$, например, x_2 или $\sim x_{15}$.
- Соседние литералы внутри одной клаузы отделяются символом \vee (означающем логическое или), окруженным одним пробелом.
- Соседние клаузы отделены символом \wedge (означающим логическое и), окруженным одним пробелом.

Суммарное количество литералов во всех клаузах не превосходит 1 000 000.

¹Байтиана забыла упомянуть, работает ли это сведение за полиномиальное время...

Формат выходных данных

Выведите количество назначений переменных x_1, \dots, x_n , при которых заданная формула верна, по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 $(x_2) \wedge (x_3 \vee \sim x_2) \wedge (x_2 \vee x_1 \vee \sim x_3)$	2

Замечание

Пояснение к примеру: формула во входе верна только для двух наборов переменных: $(0, 1, 1)$ and $(1, 1, 1)$.

Задача G. Ненокку

Имя входного файла: `nenokku.in`
Имя выходного файла: `nenokku.out`
Ограничение по времени: 4 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.

Примеры

<code>nenokku.in</code>	<code>nenokku.out</code>
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

Задача Н. Помогите, спасите!

Имя входного файла: `keepcounted.in`
Имя выходного файла: `keepcounted.out`
Ограничение по времени: 4 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

<code>keepcounted.in</code>	<code>keepcounted.out</code>
<code>aabab</code>	1 2 5 8 11
<code>atari</code>	1 3 5 9 14