Задача A. Помогите, спасите!

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
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 ($1 \leqslant N \leqslant 2 \cdot 10^5$) 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

stdin	stdout
aabab	1 2 5 8 11
atari	1 3 5 9 14

Задача B. Рефрен

Имя входного файла:	`refrain.in`
Имя выходного файла:	`refrain.out`
Ограничение по времени:	2 секунды
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 ($1 \leqslant n \leqslant 150\,000$, $1 \leqslant m \leqslant 10$).

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.

Примеры

refrain.in	refrain.out
9 3 1 2 1 2 1 3 1 2 1	9 9 1 2 1 2 1 3 1 2 1

Задача C. Пруффикс

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

Даны две строки $s = s_1 s_2 \dots s_n$ и $t = t_1 t_2 \dots t_m$. Даны несколько запросов вида (l_s, r_s, l_t, r_t), и для каждого запроса нужно посчитать число пар (x, y) таких, что

- $l_s \leqslant x \leqslant r_s$,
- $l_t \leqslant y \leqslant r_t$ и
- строка $s_x s_{x+1} \dots s_n t_1 t_2 \dots t_y$ является подстрокой s или t.

Формат входных данных

Во входных данных содержится несколько тестов. Для каждого теста:

Первая строка содержит три целых числа n, m и q, длины s, t и количество запросов соответственно ($1 \leqslant n, m, q \leqslant 5 \times 10^5$).

Вторая строка содержит строку s длины n. Третья строка содержит строку t длины m. Обе строки состоят из маленьких латинских букв.

Каждая из следующих q строк содержит четыре целых числа l_s, r_s, l_t, r_t обозначающих запрос ($1 \leqslant l_s \leqslant r_s \leqslant n$, $1 \leqslant l_t \leqslant r_t \leqslant m$).

Сумма всех значений n во всех тестах во входных данных не превышает 5×10^5.

Сумма всех значений m во всех тестах во входных данных не превышает 5×10^5.

Сумма всех значений q во всех тестах во входных данных не превышает 5×10^5.

Формат выходных данных

Выведите одно целое число для каждого запроса, являющееся ответом на запрос.

Примеры

стандартный ввод	стандартный вывод
3 3 3 aaa aaa 1 3 1 3 1 1 2 2 3 3 1 1	3 0 1

Задача D. Ненокку

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
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 полазает 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

stdin	stdout
? love ? is A Loveis ? love ? WHO A Whoareyou ? is	NO NO YES NO YES

Задача E. 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 $2 \leqslant i \leqslant K$ посчитать L_i.

Формат входных данных

В первой строке входных данных дано одно число L ($1 \leqslant L \leqslant 200\,000$) — 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 – $L_2, L_3, \ldots, L_K$.

Пример

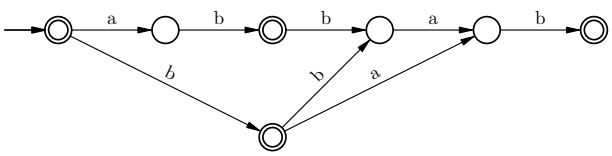
стандартный ввод	стандартный вывод
6 matter animate pattern thermal domain teammate	5 3 2 2 1

Задача F. Суффиксный автомат

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

Суффиксным автоматом для строки w называется детерминированный конечный автомат A, который допускает язык Suff(w) — множество суффиксов слова w. Например, суффиксный автомат для слова *abbab* должен допускать в точности следующие слова: $\{abbab, bbab, bab, ab, b, \varepsilon\}$. Мы также потребуем, чтобы суффиксный автомат не имел недостижимых состояний, и не было состояний, из которых не достижимы допускающие. Других ограничений, например, минимальности, накладывать не будем.

На рисунке показан суффиксный автомат для слова *abbab*.



По заданному скелету суффиксного автомата некоторого слова требуется восстановить суффиксный автомат. А именно — вам даны состояния, переходы, начальное состояние и допускающие состояния. Но пометки на ребрах удалены.

Вам следует расставить пометки на ребрах заданного суффиксного автомата, так чтобы он стал суффиксным автоматом некоторого слова w, а также найти это слово. Для простоты будем считать, что размер алфавита ничем не ограничен, вы можете использовать в качестве символов числа от 1 до k (k вы можете выбрать сами).

Формат входных данных

Первая строка входного файла содержит три целых числа: n, m и t — количество состояний, количество переходов, и количество допускающих состояний, соответственно ($2 \leqslant n \leqslant 200$, $1 \leqslant m \leqslant 1000$, $1 \leqslant t \leqslant n$). Вторая строка содержит t целых чисел — номера допускающих состояний (состояния пронумерованы с 1, начальное состояние имеет номер 1).

Следующие m строк описывают переходы: каждая строка содержит два целых числа s_i и t_i и описывает переходы из s_i в t_i.

Формат выходных данных

На первой строке выходного файла выведите два целых числа: l и k — длину слова w и размер алфавита. Используйте числа $\{1, \ldots, k\}$ как элементы алфавита. k не должно превышать m.

Вторая строка должна содержать l целых чисел — слово w.

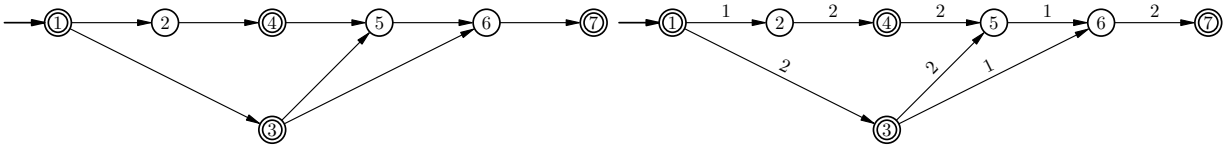
Наконец, третья строка должна содержать m целых чисел — метки на переходах скелета автомата, в том порядке, в котором они описаны во входном файле.

Гарантируется, что ответ всегда существует.

Примеры

suffix.in	suffix.out
7 8 4 1 3 4 7 1 2 1 3 2 4 3 5 3 6 4 5 5 6 6 7	5 2 1 2 2 1 2 1 2 2 2 1 2 1 2

Замечание



Задача G. Контрольное списывание

Имя входного файла:	`kthsubstr.in`
Имя выходного файла:	`kthsubstr.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Сегодня на уроке преподаватель Массивов Автомат Укконевич рассказывал своим ученикам про строки, суффиксные структуры и всё такое. Например, он рассказал им, как сравнить две строки A и B *лексикографически*. Если одна из них является префиксом другой, то более короткая будет лексикографически меньше, иначе необходимо сравнить символы стоящие на первой позиции, в которой они отличаются. Строка с меньшим по номеру в алфавите символом на данной позиции и будет лексикографически меньше.

Чтобы проверить понимание учениками нового материала, Автомат Укконевич дал им следующее задание: найти k-ю лексикографически непустую уникальную подстроку строки S.

Так как учитель знает, что Михаил В. и Роман Б. очень любят списывать у известного в узких кругах Максима И., каждый школьник получил своё число k и вынужден был обратиться к вам за помощью.

Формат входных данных

В первой строке входного файла находится строка S ($|S| \leqslant 10^5$). Вторая строка содержит число k ($1 \leqslant k \leqslant 10^{18}$) — порядковый номер запрашиваемой подстроки.

Формат выходных данных

Если ответ существует, выведите искомую подстроку строки S. В противном случае выведите её лексикографически максимальную подстроку.

Примеры

kthsubstr.in	kthsubstr.out
abacaba 10	acab
abracadabra 1000000000000000000	racadabra