

## Задача А. Рефрен

Имя входного файла: **refrain.in**  
Имя выходного файла: **refrain.out**  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

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 входного файла

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 150\,000$ ,  $1 \leq m \leq 10$ ).

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

### Формат выходного файла

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

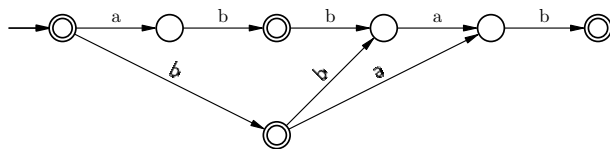
refrain.in	refrain.out
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

## Задача В. Суффиксный автомат

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

*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 допустимые. Других ограничений, например, минимальности, накладывать не будем.

На рисунке показан суффиксный автомат для слова *abbab*.



По заданному *скелету* суффиксному автомату 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 входного файла

Первая строка входного файла содержит три целых числа:  $n$ ,  $m$  и  $t$  — количество состояний, количество переходов, и количество допускающих состояний, соответственно ( $2 \leq n \leq 200$ ,  $1 \leq m \leq 1000$ ,  $1 \leq t \leq n$ ). 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 выходного файла

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

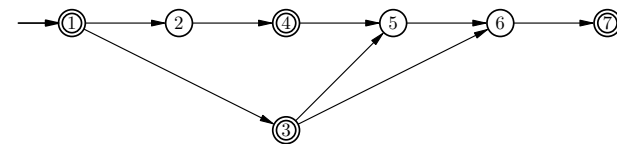
Вторая строка должна содержать  $l$  целых чисел — слово  $w$ .

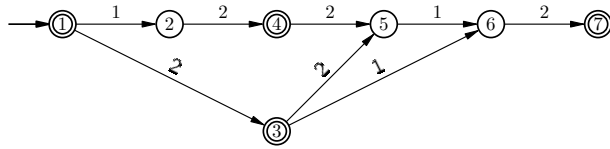
Наконец, третья строка должна содержать  $m$  целых чисел — метки на переходах скелета автомата, в том порядке, в котором они описаны во входном файле.

Гарантируется, что ответ всегда существует.

### Пример

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





### Задача С. Суффиксный пулемёт

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

Или зачёт, или автомат.

Ганнибал Ректор

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 входного файла

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 ее 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 выходного файла

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

### Пример

<code>suffix.in</code>
2 1 2
1 2
1 2 a
a
2 2 2
1 2
1 1 a
1 2 b
ab
0 0 0
<code>suffix.out</code>
Automaton 1 is a machinegun.
Automaton 2 is not a machinegun.