

# Problem A. Минимизация ДКА

Input file: `minimization.in`  
Output file: `minimization.out`  
Time limit: 2.5 секунд  
Memory limit: 256 мегабайт

Задан детерминированный конечный автомат. Требуется построить эквивалентный ему автомат с минимальным количеством состояний.

## Input

Во первой строке входного файла содержатся числа  $n$ ,  $m$  и  $k$  — количество состояний, переходов и допускающих состояний в автомате соответственно ( $1 \leq n, m \leq 1000$ ,  $1 \leq k \leq n$ ).

В следующей строке содержатся  $k$  чисел — номера допускающих состояний (состояния пронумерованы от 1 до  $n$ ).

В следующих  $m$  строках описываются переходы в формате « $a$   $b$   $c$ », где  $a$  — номер исходного состояния перехода,  $b$  — номер состояния, в которое осуществляется переход и  $c$  — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

## Output

Требуется выдать результирующий автомат в том же формате.

## Examples

<code>minimization.in</code>	<code>minimization.out</code>
2 2 2	1 1 1
1 2	1
1 2 a	1 1 a
2 2 a	

## Note

В следующей задаче требуется сделать то же самое, но с более жесткими ограничениями.

## Problem B. Быстрая минимизация ДКА

Input file: `fastminimization.in`  
Output file: `fastminimization.out`  
Time limit: 2.5 секунд  
Memory limit: 256 мегабайт

Задан детерминированный конечный автомат. Требуется построить эквивалентный ему автомат с минимальным количеством состояний.

### Input

Во первой строке входного файла содержатся числа  $n$ ,  $m$  и  $k$  — количество состояний, переходов и допускающих состояний в автомате соответственно ( $1 \leq n, m \leq 50\,000$ ,  $1 \leq k \leq n$ ).

В следующей строке содержатся  $k$  чисел — номера допускающих состояний (состояния пронумерованы от 1 до  $n$ ).

В следующих  $m$  строках описываются переходы в формате « $a$   $b$   $c$ », где  $a$  — номер исходного состояния перехода,  $b$  — номер состояния, в которое осуществляется переход и  $c$  — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

### Output

Требуется выдать результирующий автомат в том же формате.

### Examples

<code>fastminimization.in</code>	<code>fastminimization.out</code>
2 2 2	1 1 1
1 2	1
1 2 a	1 1 a
2 2 a	

## Problem C. Минимизация ДКА

Input file: `mindfa.in`  
Output file: `mindfa.out`  
Time limit: 8 секунд  
Memory limit: 256 мегабайт

Дан детерминированный конечный автомат  $A$ . Постройте детерминированный конечный автомат, принимающий тот же язык, что и  $A$ , и имеющий наименьшее возможное число состояний.

### Input

Первая строка входного файла содержит алфавит  $\Sigma$ , который является непустым подмножеством латинского алфавита (все буквы строчные).

Следующая строка содержит число  $|Q|$  — количество состояний автомата ( $1 \leq |Q| \leq 1\,000$ ).

Состояния нумеруются числами от 1 до  $|Q|$ .

Следующая строка содержит число  $q_0$  ( $1 \leq q_0 \leq |Q|$ ) — номер начального состояния, затем число  $|T|$  — количество терминальных состояний, затем  $|T|$  чисел от 1 до  $|Q|$  — номера терминальных состояний.

Следующие  $|Q|$  строк содержат по  $|\delta|$  чисел — описание функции переходов  $\delta$ . (Для каждого состояния в отдельной строке приводятся номера состояний, в которые из него ведут переходы по всем символам алфавита).

### Output

Выведите описание искомого детерминированного конечного автомата в формате, описанном выше, но без первой строки (строки с алфавитом).

### Examples

<code>mindfa.in</code>	<code>mindfa.out</code>
ab	2
5	1 1 2
1 2 2 3	2 2
2 3	1 1
1 4	
4 1	
3 2	
5 5	