

## Problem A. ASCII Area

Input file: `ascii.in`  
Output file: `ascii.out`

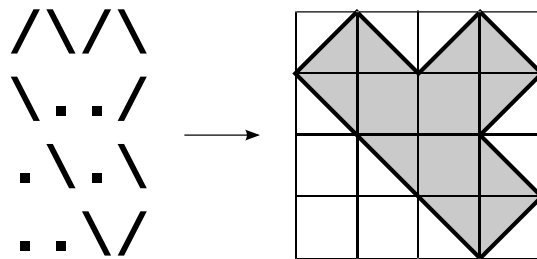
Давным-давно большинство компьютеров были оснащены видеокарточками, поддерживающими только текстовый режим. Если программист хотел нарисовать на экране картинку, ему приходилось использовать псевдографику или ASCII-художества, как это:

```

      ^  ^
      .  .
    ( 00 )
    /    \
    ( ) ( )
  
```

В этой задаче Вам даётся многоугольник, нарисованный в ASCII-арте. Ваша задача – посчитать его площадь.

Картинка состоит из символом '.', '\', '/' и '^'. Каждый символ представляет собой единичный квадрат картинки. Символ '.' символизирует пустой квадрат, символ '/' — квадрат с отрезком от нижнего левого до верхнего правого угла, а символ '\' — квадрат с отрезком от верхнего левого до нижнего правого угла.



### Input

Первая строка файла содержит целые числа  $h$  и  $w$  ( $2 \leq h, w \leq 100$ ) — высоту и ширину картинки. Следующие  $h$  строк содержат по  $w$  символов — нарисованную картинку.

Гарантируется, что на картинке находится всего один многоугольник без самопересечений и самокасаний.

### Output

Выведите единственное число — площадь многоугольника.

### Sample input and output

<code>ascii.in</code>	<code>ascii.out</code>
<pre> 4 4 /\ \ \.. / .\. \ ..\ \ </pre>	8



— ширина каждого символа,  $s_{min}$  и  $s_{max}$  ( $0 \leq s_{min} \leq s_{max} \leq 30$ ) — минимальная и максимальная ширина пробела в пикселях.

Следующие  $m$  строк ввода содержат описание шрифта. Каждая строка содержит  $t(k+3) - 1$  символов, где  $t$  ( $1 \leq t \leq 26$ ) — количество латинских букв, которые используются в этом шрифте. Таблица с  $m$  строками и  $t(k+3) - 1$  столбцами на этих  $m$  строках создана из  $m \times k$  таблиц символов ‘.’ and ‘\*’, определяющими шрифт для больших латинских букв от A до Z. Соответствующая буква написана перед её таблицей в первой строке (см. пример ввода). Первая из этих  $m$  строк использует  $2t - 1$  пробелов как разделители, последующие — по  $3t - 1$  пробелов. Буквы могут быть заданы не в алфавитном порядке, но каждая буква определена хотя бы 1 раз.

Пробельный символ в каждом шрифте — это таблица  $m \times k$  состоящая только из ‘.’. Он является обычным символом и может отстоять от соседних на расстояние между  $s_{min}$  и  $s_{max}$ .

Следующая строка содержит текст, который отображается на табличке в данный момент. Эта строка содержит  $c_{cur}$  символов ( $1 \leq c_{cur} \leq 30$ ) — большие латинские буквы от A до Z и пробелы. Нет начальных или конечных пробелов.

Следующая строка содержит  $c_{cur}$  неотрицательных чисел. Каждое число определяет количество пустых столбцов перед соответствующим символом или пробелом отображаемой в данный момент надписи. Первое число — это расстояние от левой границы надписи до первой буквы, второе число — расстояние от первой буквы до второй буквы или пробела и так далее. Вся строка уместится на надписи. Расстояния в исходном тексте могут не подчиняться ограничениям  $s_{min}$  и  $s_{max}$ .

Следующая строка содержит новый текст. В ней  $c_{new}$  символов ( $1 \leq c_{new} \leq 30$ ) — больших латинских букв от A до Z или пробелов без лидирующих и завершающих пробелов.

Все буквы, которые используются, определены в описании шрифта.

## Output

В выходной файл выведите единственную строку с  $c_{new}$  числами, показывающую оптимальные расстояния для нового текста. Первое число — это расстояние от левой границы надписи до первой буквы, второе число — расстояние от первой буквы до второй буквы или пробела и так далее. Первое число должно быть неотрицательно, остальные числа должны лежать в промежутке от  $s_{min}$  до  $s_{max}$  включительно. Текст должен уместиться на надпись. Всегда есть хотя бы один способ разместить надпись, удовлетворяя указанным требованиям. Если есть несколько оптимальных ответов, выведите любой из них.

## Sample input and output

caption.in	
5 53 5 1 2	
A .*. . C .**** E ***** I .*. . M *.*.* N *.*.* P ****. R ****.	
.*. . *.*.* *.*.* *.*.* *.*.* *.*.* *.*.* *.*.*	
*.*.* *.*.* *.*.* *.*.* *.*.* *.*.* *.*.* *.*.*	
***** *.*.* *.*.* *.*.* *.*.* *.*.* *.*.* *.*.*	
*.*.* *.*.* *.*.* *.*.* *.*.* *.*.* *.*.* *.*.*	
ACM ICPC	
3 1 1 1 1 1 1 1	
NEERC	
caption.out	
19 2 2 1 1	

## Problem C. Eve

Input file:       eve.in  
Output file:      eve.out

Митохондриальная ДНК это молекула дезоксирибонуклеиновой кислоты, которая содержится в митохондриях в клетках нашего организма.

Митохондриальная ДНК обычно передаётся ребёнку от матери. Из-за этого, можно ввести понятие “Митохондриальная Ева”, которое будет значить наиболее молодую общую мать всей популяции.

Митохондриальная Ева всех жителей Земли жила примерно 200 000 лет назад, скорее всего, в Южной Африке.

В этой задаче Вам дана популяция некоторого эукариотического и анизогамного вида. За популяцией следили некоторое время, и все рождения и смерти были записаны. Для некоторых существ в популяции была записана их Митохондриальная ДНК. Считается, что каждое существо получает Митохондриальную ДНК от матери без мутаций.

Ваша задача – найти, у всех ли ныне живущих существ одинаковая Митохондриальная ДНК.

### Input

Первая строка содержит число  $n$  ( $1 \leq n \leq 100\,000$ ), количество существ, которые были живы на момент начала периода наблюдения. Их номера – числа от 1 до  $n$ .

Следующие  $n$  строк содержат по одному символу.  $i$ -ая строчка описывает пол существа с номером  $i$ . ‘M’ значит мужской, ‘F’ – женский.

Следующая строка содержит число  $m$  ( $0 \leq m \leq 100\,000$ ), общее количество рождений и смертей за период.

Следующие  $m$  строк содержат описания событий рождения или смерти в хронологическом порядке.

Рождение описывается тремя частями – номером отца, номером матери и символом ‘M’ или ‘F’ – полом ребёнка. Ребёнку даётся минимальный номер, который не использовался ранее.

Смерть описывается одним отрицательным числом, чей модуль равен номеру существа, которое погибло.

Следующая строка содержит число  $k$  ( $0 \leq k \leq n + m$ ), количество записанных Митохондриальных ДНК.

Следующие  $k$  строк содержат по два разделенных пробелом целых числа – номер существа, чья Митохондриальная ДНК была записана и число – уникальный идентификатор этой Митохондриальной ДНК. Идентификаторы совпадают, только если Митохондриальные ДНК существ совпадают. Все идентификаторы – числа от 1 до  $10^9$ .

Все данные непротиворечивы. ДНК каждого существа была записана хотя бы один раз. Хотя бы одно существо живо в конце периода наблюдения.

### Output

Выходной файл должен соержать одно слово:

- YES – если можно доказать, что у всех оставшихся существ одинаковая Митохондриальная ДНК.
- NO – если можно доказать, что не у всех оставшихся существ одинаковая Митохондриальная ДНК.
- POSSIBLY – в противном случае.

## Sample input and output

eve.in	eve.out
<pre> 4 M F M F 12 3 4 F 1 2 M 1 2 M 3 4 F -3 -2 -4 -1 6 5 M 7 5 F -7 -6 0 </pre>	<pre> YES </pre>
<pre> 3 F F M 3 3 2 M 3 1 F -3 2 4 100500 5 100500 </pre>	<pre> YES </pre>
<pre> 3 M F M 2 1 2 M 3 2 F 0 </pre>	<pre> POSSIBLY </pre>
<pre> 2 M F 2 1 2 M -2 2 1 2011 2 2012 </pre>	<pre> NO </pre>