
Задача А. Национальное достояние

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 512 мегабайт |

Как вы все знаете, в стране Книгерии находится самая большая в мире библиотека. В ней есть десятки тысяч книг, написанных в самые разные периоды человеческой истории.

Прогресс не стоит на месте, и со временем руководство библиотеки приняло решение о создании электронной базы, содержащей названия всех книг, находящихся в библиотеке. Денис — программист, которому была поручена работа по созданию базы данных и по заведению в ней записей, отвечающих всем книгам в библиотеке.

Как типичный начинающий программист, вместо использования любого из имеющихся решений для этой задачи Денис решил написать свою базу данных с нуля. После нескольких месяцев кропотливого труда его база данных заработала, и он даже смог вбить в неё названия всех книг, но на презентации результата своих трудов он с удивлением узнал, что книги в базе данных должны следовать в лексикографическом порядке их названий. База данных Дениса ещё не поддерживает такую сложную процедуру, как сортировка, а перебивать все названия с нуля Денису очень не хочется, поэтому он решил схитрить.

Книгерийский алфавит настолько большой, что его буквы обозначаются целыми положительными числами. Каждая буква алфавита бывает строчной и прописной, прописная версия буквы, соответствующей числу x , обозначается как x' . Кодировка BSCII, используемая повсеместно в Книгерии, устроена таким образом, что прописные буквы упорядочиваются в порядке увеличения числа, обозначающего букву, аналогично упорядочиваются строчные буквы, но при этом все прописные буквы алфавита следуют **до** всех строчных букв алфавита. Например, верны следующие утверждения: $2 < 3$, $2' < 3'$, $3' < 2$.

Слово x_1, x_2, \dots, x_a не превосходит слова y_1, y_2, \dots, y_b в *лексикографическом* порядке если верно одно из двух условий:

- либо $a \leq b$ и $x_1 = y_1, \dots, x_a = y_a$, то есть первое слово является префиксом второго;
- либо есть такая позиция $1 \leq j \leq \min(a, b)$, что $x_1 = y_1, \dots, x_{j-1} = y_{j-1}$ и $x_j < y_j$, то есть, в первой позиции, в которой слова отличаются, в первом слове стоит меньшая буква.

Как следствие, слово «3' 7 5» следует в лексикографическом порядке до слова «2 4' 6». Говорят, что слова в последовательности следуют в лексикографическом порядке, если каждое слово последовательности лексикографически не превосходит следующего слова в последовательности.

На данный момент названия всех книг записаны строчными буквами. Денис хочет поменять некоторые буквы на прописные (такой процесс называется *капитализацией*) таким образом, чтобы названия книг стали следовать в лексикографическом порядке. Однако в процессе реализации этого плана возникла проблема — из-за неаккуратного устройства базы данных Денис не может капитализировать одну букву в конкретном названии. Единственная операция, которую он может проделать с буквой, это капитализировать все вхождения данной буквы в названия **всех** книг. Такую операцию можно проделать любое количество раз с произвольными буквами книгерийского алфавита.

Помогите Денису выбрать, какие буквы надо капитализировать, чтобы заданный порядок следования названий книг стал лексикографическим, либо определите, что это невозможно.

Обратите внимание, в библиотеке могут храниться книги с **одинаковыми** названиями.

Формат входных данных

В первой строке даются числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — количество книг в книгерийской библиотеке и количество букв в книгерийском алфавите соответственно. Буквы книгерийского алфавита обозначаются целыми числами от 1 до m .

Каждая из последующих n строк содержит описание книги в формате $l_i, s_{i,1}, s_{i,2}, \dots, s_{i,l_i}$ ($1 \leq l_i \leq 100\,000$, $1 \leq s_{i,j} \leq m$), где l_i обозначает длину названия книги, а $s_{i,j}$ задаёт последовательность букв в названии книги.

Гарантируется, что суммарная длина названий книг не превышает 100 000.

Формат выходных данных

В первой строке выведите «Yes» (без кавычек), если возможно капитализировать некоторое подмножество букв таким образом, чтобы последовательность названий книг стала отсортированной лексикографически. Иначе выведите «No» (без кавычек).

Если требуется, во второй строке выведите k — количество букв, которые надо капитализировать, а в третьей строке выведите k различных чисел — номера этих букв. Обратите внимание, минимизировать значение k **не требуется**.

Номера букв можно выводить в любом порядке. Если существует несколько возможных ответов, выведите любой из них.

Примеры

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 4 3 1 2 1 1 3 1 3 2 2 1 1 | Yes 2 2 3 |
| 6 5 2 1 2 2 1 2 3 1 2 3 2 1 5 2 4 4 2 4 4 | Yes 0 |
| 4 3 4 3 2 2 1 3 1 1 3 3 2 3 3 2 3 1 | No |

Замечание

В первом примере названия книг после капитализации букв 2 и 3 выглядят следующим образом:

- 2'
- 1
- 1 3' 2'
- 1 1

Верно соотношение $2' < 1$, поэтому название первой книги лексикографически не превосходит названия второй. Название второй книги является префиксом названия третьей, следовательно они идут в лексикографическом порядке. Так как у названий третьей и четвертой книги первая буква совпадает, а $3' < 1$, то название третьей книги не превосходит названия четвертой книги лексикографически.

Во втором примере названия книг уже расположены в лексикографическом порядке, а значит можно не капитализировать буквы.

В третьем примере не существует набора букв, путём капитализации которых можно добиться, чтобы названия книг были расположены в лексикографическом порядке.