

## Console. Поиск набора образцов 1, 2

Имя входного файла: `console.in`  
Имя выходного файла: `console.out`

Напишите программу, которая для каждой строки из заданного набора  $S$  проверяет, верно ли, что она содержит как подстроку одну из строк из набора  $T$ .

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  — количество строк в наборе  $T$ . Каждая из следующих  $n$  строк содержит непустую строку длины не более 80-ти символов.

Оставшаяся часть файла содержит строки из набора  $S$ . Каждая строка состоит из ASCII символов с кодами от 32 до 126 включительно. Строка может быть пустой и ее длина не превышает 250-ти символов.

Гарантируется, что размер входного файла не превышает 1 Мбайт.

### Варианты

Console1:  $n \leq 100$ .

Console2:  $n \leq 1000$ .

### Формат выходного файла

В выходной файл выведите все строки из набора  $S$  (в том порядке, в котором они находятся во входном файле), содержащие как подстроку по крайней мере одну строку из набора  $T$ .

### Пример

<code>console.in</code>	<code>console.out</code>
3 gr sud abc lksh sudislavl kostroma summer group b	sudislavl group b

## Kthstr. К-ая строка

Имя входного файла: `kthstr.in`  
Имя выходного файла: `kthstr.out`

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку  $S$ ;
- найти в словаре  $k$ -ую строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $N$  — количество команд ( $N \leq 10^5$ ). Последующие  $N$  строк содержат по одной команде каждая.

Команда записывается в виде числа  $k$ , либо строки  $S$ , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе  $k$ -ой строки, она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает  $10^5$ .

### Формат выходного файла

Для каждого числового запроса  $k$  выходной файл должен содержать  $k$ -ую в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что сумма длин строк в выходном файле не превышает  $10^5$ .

### Пример

<code>kthstr.in</code>	<code>kthstr.out</code>
7 pushkin lermontov tolstoy gogol gorkiy 5 1	tolstoy gogol

## Yandex. Яндекс

Имя входного файла: `yandex.in`  
Имя выходного файла: `yandex.out`

Яндекс работает в одной очень известной компании. Его работа не очень сложна, но требует много времени. В основном Яндекс ищет некоторые данные в одной книге и переписывает их в другую. Яндекса не особенно волнует, кому нужны результаты его работы, главное — что за работу хорошо платят. Яндекс пришел в эту компанию не так давно, поэтому он трудится добросовестно и очень устает к концу дня. К концу дня для него все символы в книге сливаются, так что все эти ценные данные — это одна длинная строка, но он должен еще работать с ним и работать дальше и дальше... Может, босс заметит, как тщательно работает Яндекс, и повысит его...

Но... О, нет... Пока Яндекс мечтал, он забыл, что он должен был смотреть в первой книге... После перерыва и чашки чая «Липтон» он кое-что вспомнил. Во-первых, он вспомнил, что он должен был искать какую-то строку в первой книге. Во-вторых, он вспомнил, что во вторую книгу он должен был выписывать позиции, в которых встречалась эта строка, и что он уже выписал их все.

### Формат входного файла

Во входном файле содержатся несколько тестов. Описание каждого теста начинается с натуральных чисел  $n$  ( $1 \leq n \leq 1\,000\,000$ ) — количество символов в первой книге — и  $k$  ( $1 \leq k \leq n$ ) — количество позиций, в которых Яндекс уже нашел вхождения искомой строки в текст (т.е. количество чисел во второй книге). На второй строке описания теста находится текст из первой книги — последовательность символов с ASCII-кодами большими, чем 64. Третья строка описания теста содержит  $k$  номеров позиций, которые были записаны во второй книге.

Строка с  $n = k = 0$  обозначает конец тестов; этот тест и все данные после него не должны быть обработаны.

### Формат выходного файла

Для каждого теста выведите в выходной файл одну строку. Если существует строка, которая входит в текст в тех и только в тех позициях, что указаны во второй книге, выведите одну строку `Correct. Length =  $x..y$ .`, где  $x$  и  $y$  — минимально и максимально возможная длина искомой строки. Если решения не существует, выведите в выходной файл одну строку `Mistake.`

## Пример

	<code>yandex.in</code>	<code>yandex.out</code>
5	2	Correct. Length = 2..3.
ababa		Correct. Length = 1..1.
1	3	Mistake.
1	2	
a		
1	1	
2	2	
ab		
1	2	
0	0	