

## Задача А. Суффиксный путь

Имя входного файла:	sufpath.in
Имя выходного файла:	sufpath.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В одной супер-секретной лаборатории «Кимод йытысед» было разработано супер-секретное устройство. Синдикату «Черное солнце» удалось выяснить, что устройство имело супер-секретное название «Тамотва 173». Лучшие умы пытались расшифровать его, но так ничего и не удавалось. Пробовали и шифр цезаря, и сдвиг, и RSA. В конце концов, синдикат обратился за помощью в Весенний Гуманитарный Детский сад (ВГД). Как оказалось, это — также супер-секретная организация, которая занимается подготовкой элитного отряда дворников. Они профессионально умеют складывать мусор на самых видных местах, подметать так, что просыпается весь район и также немного увлекаются философией. Разумеется, они мгновенно поняли, что в названии зашифровано ни что иное, как основная часть этого устройства! Еще они выяснили, что это устройство умеет обрабатывать только файлы размера не более, чем шестьдесят мегабайт.

После этого синдикат обратился за помощью к лучшим друзьям ВГД, Еженедельному Женевскому Завтраку (ЕЖЗ). Они также специализируется на шифрах: красят заборы, моют полы, а также занимаются модернизацией и инновациями. ЕЖЗ сообщили, что это устройство — их профиль, оно идеально сочетается с их идеологией. Более формально, устройство также умеет выполнять модернизацию. К сожалению, с инновациями может справиться только ЕЖЗ, поэтому устройство не умеет их придумывать, а может лишь проверять, является ли некоторая инновация действительно революционной инновацией. Как оказалось, внутри устройства хранится строка, состоящая из маленьких латинских букв. Модернизация состоит в том, чтобы к хранящейся строке дописать маленькую латинскую букву. ЕЖЗ хотели сообщить более подробную информацию, но после фразы, что ВГД ошиблись в подсчете максимального размера обрабатываемого файла в тридцать раз, они были в непригодном для общения состоянии: у большинства отвалилась челюсть.

Синдикат обратился к своим последним друзьям: команде Инноваций и Культа Лени (ИКЛ). Они сразу объяснили, что строка является для устройства инновационной, если она является частью хранящейся в устройстве строки. Но всем сразу стало ясно: здесь есть какой-то подвох! И снова лучшие умы стали биться над этой задачей. Перепробовали всё: и кричать, и танцевать, и петь песенки, и бить баклуши. Но так им и не удавалось понять, что же происходит на самом деле. Наконец, директор ИКЛ пришёл к директору ЕЖЗ, и они, вместе с директором ВГД и мокренькой кисонькой, поняли, что строка должна быть не просто частью, а, будучи развернутой, должна являться префиксом развернутой строки, хранящейся в устройстве! Более того, она также должна состоять из маленьких латинских букв. После этого все вместе они пошли спать.

Из более достоверных источников (а именно Мадагаскарский Национальный Отряд Профессиональных Супер-агентов - МНОП) стало известно, что тот, у кого окажется данное устройство, получит неограниченную власть над миром. Ваша задача кристально ясна: реализуйте данное устройство.

### Формат входного файла

В первой строке входного файла содержится число  $M$  — количество выполняемых операций. Далее в  $M$  строках содержатся описания операций: либо  $+$   $s$  для модернизации, либо  $?$   $s$  для проверки инновационности. Гарантируется, что устройство сможет обработать входной файл.

### Формат выходного файла

Для каждого запроса проверки выведите «YES», если строка является инновационной, и «NO» в противном случае.

**Примеры**

sufpath.in	sufpath.out
2 + a ? a	YES

## Задача В. Цензура

Имя входного файла: `censored.in`  
Имя выходного файла: `censored.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Посчитайте, сколько строк над алфавитом из  $n$  символов длины  $m$  не содержат ни одной подстроки из заданного множества “запрещенных” строк.

### Формат входного файла

В первой строке написаны целые числа  $n$  ( $1 \leq n \leq 100$ ) — количество символов в алфавите,  $m$  ( $1 \leq m \leq 100$ ) — длина искомым строк и  $p$  ( $0 \leq p \leq 10$ ) — количество “запрещенных” подстрок. Следующая строка содержит  $n$  символов с кодами больше 32 — буквы алфавита. Далее идет  $p$  “запрещенных” строк, длины которых не превосходят  $\min(m, 10)$  символов. Строки целиком состоят из символов алфавита.

### Формат выходного файла

В первой строке выведите ответ на задачу.

### Примеры

<code>censored.in</code>	<code>censored.out</code>
2 3 1 ab bb	5

## Задача С. Период строки

Имя входного файла: `period.in`  
Имя выходного файла: `period.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $s$ . Требуется найти минимальную по длине строку  $t$ , такую что  $s$  представима в виде конкатенации одной или нескольких строк  $t$ .

### Формат входного файла

Первая строка входного файла содержит  $s$  ( $1 \leq |s| \leq 5 \cdot 10^6$ ). Строка состоит из букв латинского алфавита.

### Формат выходного файла

Выведите длину искомой строки  $t$ .

### Примеры

<code>period.in</code>	<code>period.out</code>
abcabcabc	3

## Задача D. KINA is not abbreviation

Имя входного файла:	kina.in
Имя выходного файла:	kina.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

При оперировании новыми терминами, состоящими из нескольких слов, полезно вводить аббревиатуры. *Аббревиатурой* называется слово, состоящее из первых букв нескольких подряд идущих слов.

Аббревиатура называется *однозначной*, если выполнены следующие два условия:

- Она соответствует в точности одной последовательности слов заданного текста (хотя эта последовательность слов может встречаться в тексте более одного раза).
- Она не встречается в тексте сама по себе в качестве слова.

Например, в тексте «**A recursive acronym KINA means "KINA is not abbreviation"**», строки «**ARA**» и «**К**» являются однозначными аббревиатурами, строки «**A**» и «**KINA**» являются неоднозначными аббревиатурами, а строки «**RAA**» и «**KNA**» не являются аббревиатурами.

Чтобы ввести аббревиатуру в тексте, ее пишут в скобках сразу после первого появления последовательности слов, которой эта аббревиатура соответствует. Последующие вхождения этой последовательности слов могут быть заменены аббревиатурой. Например, в тексте, приведенном выше, после введения аббревиатуры «**К**» получается следующее: «**A recursive acronym KINA (K) means "K is not abbreviation"**».

Если несколько вхождений последовательности слов пересекаются, только одно из них можно заменить аббревиатурой. Словами считаются последовательности подряд идущих латинских букв, разделенных не латинскими буквами. Слова сравниваются без учета регистра. Например, «**i18n**» является вхождением последовательности слов «**I n**».

*Эффективностью* аббревиатуры называется уменьшение количества букв в тексте после ее введения. В расчет берутся только латинские буквы, пробелы, скобки и прочие неалфавитные символы не считаются.

Вам дан текст. Найдите однозначную аббревиатуру с максимальной эффективностью.

### Формат входного файла

Входной файл содержит текст, максимум из 4000 символов. Текст содержит только символы с кодами от 32 (пробел) до 126 («~»), 13 (возврат каретки), и 10 (перевод строки).

### Формат выходного файла

Если в тексте нет однозначной аббревиатуры с положительной эффективностью, выведите в выходной файл единственное число 0.

В противном случае в первую строку выходного файла выведите эффективность оптимальной аббревиатуры. Во второй строке выведите саму аббревиатуру. Если существует несколько однозначных аббревиатур с максимальной эффективностью, выведите любую.

## Примеры

kina.in	kina.out
This problem name is "KINA is not abbreviation". Once again: KINA is not abbreviation.	11 INA
To be or not to be: that is the question.	0
Here is the chorus of the song "Jingle Bells": Jingle bells, jingle bells, Jingle all the way; Oh what fun it is to ride In a one-horse open sleigh.	16 JB

## Note

В первом примере оптимальными аббревиатурами являются «NA» и «INA».

В третьем примере оптимальными аббревиатурами являются «JB» и «BJ».

## Задача Е. К-я строка

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

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку  $S$ ;
- найти в словаре  $k$ -ю 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 либо в виде числа  $k$ , либо в виде строки  $S$ , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе  $k$ -й 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  $k$ -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает  $10^5$ .

### Примеры

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