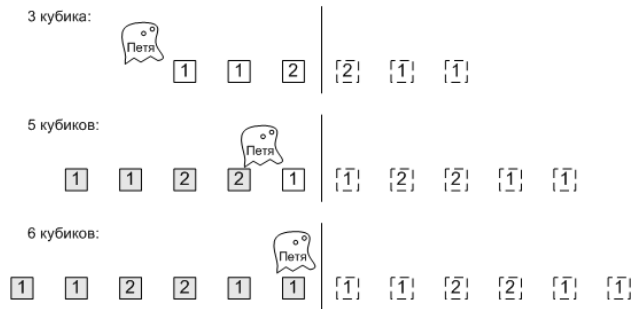


Задача А. Кубики

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

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале.



Помогите Пете! Выясните, сколько у него может быть кубиков. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Формат входного файла

Первая строка входного файла содержит два целых числа: N ($1 \leq N \leq 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики, — M ($1 \leq M \leq 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Формат выходного файла

В выходной файл выведите в порядке возрастания все такие K , что у Пети может быть K кубиков.

Примеры

| <code>cubes.in</code> | <code>cubes.out</code> |
|-----------------------|------------------------|
| 6 2 1 1 2 2 1 1 | 3 5 6 |

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. 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 «K» 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 «K» 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 D. Фабрика палиндромов

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

Рассмотрим произвольную строку g . Будем называть эту строку *генератором палиндромов*. Множество палиндромов $P(g)$, которые генерируются этой строкой определяется следующим образом.

Пусть длина строки равна n . Для всех i от 1 до n в $P(g)$ включаются строки $g[1..i]g[1..i]^r$ и $g[1..i]g[1..i-1]^r$, где α^r означает α , записанную в обратном порядке.

Например, если $g = \langle \text{olymp} \rangle$, то $P(g) = \{ \langle \text{oo} \rangle, \langle \text{o} \rangle, \langle \text{ollo} \rangle, \langle \text{olo} \rangle, \langle \text{olyulo} \rangle, \langle \text{olylo} \rangle, \langle \text{olyummylo} \rangle, \langle \text{olyulyo} \rangle, \langle \text{olyumppmylo} \rangle, \langle \text{olyumrmylo} \rangle \}$.

Для заданного генератора палиндромов g и строки s требуется найти количество вхождений строк из $P(g)$ в s как подстроки. А именно, требуется найти количество таких пар (i, j) , что $s[i..j] \in P(g)$.

Формат входного файла

Первая строка входного файла содержит строку g . Вторая строка входного файла содержит s . Обе строки непусты и имеют длину не больше 100 000 символов.

Формат выходного файла

Выведите в выходной файл количество таких пар (i, j) , что $s[i..j] \in P(g)$.

Примеры

| palin.in | palin.out |
|------------------------------------|-----------|
| olymp olleolleolympmyolylyomylo | 7 |