

Задача А. Послание к инопланетянам

Входной файл: `aliens.in`
Выходной файл: `aliens.out`
Ограничение по времени: 2 секунды
Ограничение памяти: 64 мегабайта

Программист Вася хочет отправить послание длины k к инопланетянам. Для этого он взял текст одной из своих любимых песен, и хочет в него вставить пробелы таким образом, чтобы строчка, составленная из n -го, $2n$ -го, \dots , kn -го символа образовывала заданное послание ($(k+1)n$ -й и дальнейшие символы не входят в послание). Пробелы можно вставлять только на тех местах, где уже встречается хотя бы один пробел.

Все символы с кодом 13 (CR) во входном файле игнорируются, а символ с кодом 10 (LF) считается эквивалентным символу с кодом 32 (пробелу). Таким образом, один перевод строки во всей задаче считается полностью эквивалентным одному пробелу.

Естественно, так как каждый символ, отправляемый инопланетянам, стоит денег, то Вася хочет отправить послание минимально возможной длины (перевод строки опять же считается одним символом, так как на посылающей машине стоит UNIX). Как ему это сделать?

Описание входного файла

В первой строке входного файла содержится натуральное число n , меньшее 250. Во второй строке содержится послание, которое должно быть закодировано. Начиная с третьей строки идет сам текст песни, содержащий только символы с кодами 32..255, а также последовательности из двух символов 13 10 (т.е. перевод строки), считающиеся эквивалентными одному пробелу.

Длина текста песни не превосходит 1000 символов (перевод строки считается за один символ), а длина послания — 50 символов. Послание может содержать пробелы, но не переводы строки (при этом разрешается сопоставить пробелу в послании перевод строки в тексте). Символы, отличающиеся регистром написания, считаются различными.

Описание выходного файла

В первой строке выходного файла выведите минимально возможное число вставляемых пробелов. Начиная со второй строки, необходимо вывести текст минимальной длины со вставленными пробелами, в котором можно прочесть заданное послание. Переводы строки рассматриваются проверяющей программой аналогично правилам, применимым ко входному файлу (один перевод строки считается эквивалентным одному пробелу).

Если в текст песни невозможно вставить пробелы таким образом, то выведите в выходной файл единственное слово IMPOSSIBLE.

Для удобства в примере все пробелы заменены на звездочки. Ваше решение должно выводить настоящие пробелы (то есть символы с кодом 32)!

Пример

<code>aliens.in</code>	<code>aliens.out</code>
18 привет Уронили мишку на пол, Оторвали мишке лапу, Все равно его не брошу, Потому что он хороший!	32 Уронили*мишку*на*пол,***** Оторвали*****мишке**лапу, Все*равно*****его*не**брошу, Потому*что*он*хороший!

Задача В. Уголок

Входной файл: `angle.in`
Выходной файл: `angle.out`
Ограничение по времени: 2 секунды
Ограничение памяти: 64 мегабайта

Из клетчатого листа по линиям сетки вырезан уголок — фигура, которая получается, если из прямоугольника $a_1 \times a_2$ удалить прямоугольник $b_1 \times b_2$ так, что:

- эти прямоугольники имеют общую угловую клетку;
- сторона длины a_1 первого из них параллельна стороне длины b_1 второго.

Назовем *маленьким* уголок, у которого $a_1 = a_2 = 2$, $b_1 = b_2 = 1$. Требуется найти количество способов разрезать по линиям сетки заданный уголок на маленькие уголки.

Описание входного файла

Входной файл содержит целые числа a_1 , b_1 , a_2 и b_2 ($0 \leq b_1 \leq a_1 \leq 13$, $a_1 - b_1 \leq 6$, $0 \leq b_2 \leq a_2 \leq 13$, $0 \leq a_2 - b_2 \leq 6$).

Описание выходного файла

Выведите искомое число способов.

Пример

angle.in	angle.out
6 3 5 2	16

Задача С. Циклические строки

Входной файл: *нет*
Выходной файл: *нет*
Ограничение по времени: *нет*
Ограничение памяти: *нет*

Циклическая строка представляет собой обычную строку с точностью до циклических сдвигов. Например, **омах** и **ахом** — одинаковые циклические строки, поскольку получаются друг из друга циклическим сдвигом. Напротив, **andrew** и **werdna** — разные циклические строки.

Скажем, что циклическая строка ω *содержит строку* α в качестве подстроки, если α содержится в строке, получаемой повторением ω бесконечное число раз. Например, циклическая строка **омах** содержит строки **хо** и **хом** в качестве подстрок, а строку **ам** — не содержит.

Вам заданы натуральные числа n и k ($1 \leq n \leq 40$, $1 \leq k \leq 26$) и непустая строка α (не циклическая!). Рассмотрим алфавит \mathcal{A} , состоящий из первых k маленьких букв латинского алфавита. Сколько существует циклических строк длины n над этим алфавитом, содержащих α в качестве подстроки?

Описание входного файла

Вам представлен набор из 14 тестов. Файлы, содержащие тесты, называются **сyclic.##**. Таким образом, файлы имеют названия **сyclic.01**, **сyclic.02**, ..., **сyclic.14**. В первой строке входного файла заданы числа n и k . Во второй строке задана строка α . Все символы строки α принадлежат алфавиту \mathcal{A} .

Описание выходного файла

Выходные файлы следует положить в каталог **p30_<id>_с**, где **<id>** — ваш идентификатор, и назвать их **сyclic.##**. В выходной файл выведите единственное число — количество искомых циклических строк.

Пример

входной файл	выходной файл
4 2 аа	3