

Вариант 1

Во всех тестах все строки входного файла завершаются символом переноса строки.

A. Буквы, которых нет

Дан файл, содержащий строки из заглавных латинских букв.

Требуется вывести строку, состоящую из заглавных букв латинского алфавита, не встречающихся во входном файле и не содержащей никаких других символов. Буквы в этой строке должны быть упорядочены по алфавиту и не повторяться. Если в файле встречаются все буквы, требуется вывести слово NONE.

| Input | Output |
|----------------------------------|--------|
| ABCDEFGH IJKLMNOPRQ RSTUVW | XYZ |

B. Самые частые буквы

Дан файл, содержащий строки из заглавных латинских букв. Гарантируется, что в каждой строке файла содержится хотя бы одна буква.

Для каждой строки входного файла вывести букву, встречающуюся чаще всего в этой строке. Если таких букв несколько, вывести ту, которая в строке **встречается раньше**.

| Input | Output |
|---------------|--------|
| BAB DCZZAA | B Z |

C. Две пустые строки

Дан файл, содержащий произвольные символы и ровно две строки, содержащие только символ переноса строки (т.е. ни имеющие других символов).

Подсчитать количество символов в файле между этими пустыми строками. Символы переноса строки не учитывать.

| Input | Output |
|----------------------------------|--------|
| ABCD DFG FR ZXR | 5 |

D. Файлы и расширения

Дан текст, содержащий произвольные символы. Среди прочего он содержит имена файлов в следующем формате: <имя>.<расширение>, где <имя> и <расширение> это непустые последовательности латинских букв. Названием файла считается только указанная последовательность символов, ограниченная слева и справа пробелом или концом/началом строки. На самом деле файлы, конечно, могут содержать произвольное количество точек или цифр, но в этой задаче мы ограничимся только названиями указанного рода.

Требуется в выходной файл вывести в точности содержимое исходного файла, поменяв все расширения файлов `bmp` на `jpg` (регистр учитывается, т.е. расширения вида `bMp` вас не интересуют).

Обратите внимание — все пробелы должны остаться на месте, т.е. `f.readline().split()` здесь неуместен, потому что все повторяющиеся пробелы он «съест». Это тот редкий случай, когда `split()` используется с явным указанием разделителя-пробела: `split(' ')`.

| Input | Output |
|---|---|
| test.bmp test.png.bmp try3.bmp g.jpg try.bmp | test.jpg test.png.bmp try3.bmp g.jpg try.jpg |

Комментарии к примеру:

- `test.png.bmp` не является именем файла, т.к. содержит две точки.
- `try3.bmp` не является именем файла, т.к. слева от точки встречаются не только символы-буквы.

Е. Блоки и заголовки

Входной файл содержит некоторое количество блоков следующего вида:

>строка-заголовок

строка-текст

строка-текст

...

строка-текст

Каждый блок начинается со строки-заголовка. Строка-заголовок начинается с символа '>' и содержит не менее 2 символов. После строки-заголовка следует некоторое количество строк (не менее 1), содержащих только заглавные латинские буквы и символы переноса строк. Гарантируется, что строки-заголовки попарно различны.

Требуется вывести строки-заголовки (без символа '>') в порядке убывания количества строк в соответствующем этому заголовку блоке. Если блоков с одинаковым количеством строк несколько, упорядочить их заголовки лексикографически.

| Input | Output |
|--|----------------------------|
| >TITLE3 SFG ETYJ | TITLE2 TITLE1 TITLE3 |
| >TITLE2 VKJHJVHJB HJJHVJHV LHJH LJHB | |
| >TITLE1 KJHK A | |

Ф. Биграммы

Дан файл, содержащий пробелы, заглавные латинские буквы и символы переноса строки.

Биграммой называется последовательность двух рядом стоящих букв в тексте. Требуется вычислить количество вхождений в текст всех его биграмм и вывести эти значения в порядке убывания. Биграммы с одинаковым числом вхождений вывести в лексикографическом порядке по возрастанию.

Гарантируется, что во входном файле есть хотя бы одна биграмма.

| Input | Output |
|----------------------------|--|
| ABRACADABRA Z | AB 2 BR 2 RA 2 AC 1 AD 1 CA 1 DA 1 |
| ON A B C D E F X Y Z | ON 1 |

Г. Шарик

Есть N шариков, висящих в воздухе вдоль одной линии на разной высоте слева направо.

Амур стреляет в шарик слева, запуская стрелу на некоторой высоте. Стрела летит слева направо до тех пор, пока не попадёт в шарик. В тот момент, когда она его касается, шарик лопается, а стрела летит дальше на высоте, уменьшенной на 1. То есть, если стрела летела на высоте H , то после столкновения она будет лететь на высоте $H - 1$.

Цель Амура — сбить все шарик, используя минимальное количество стрел.

В первой строке записано одно натуральное число N ($1 \leq N \leq 10^6$). Во второй строке записано N натуральных чисел H_i . Каждое число H_i ($1 \leq H_i \leq 10^{18}$) обозначает высоту, на которой висит i -й шарик в порядке слева направо.

В единственной строке выведите одно целое число — минимальное количество выстрелов, необходимое для того чтобы сбить все шарик.

| Input | Output |
|----------------|--------|
| 5 2 1 5 4 3 | 2 |
| 5 1 2 3 4 5 | 5 |
| 5 4 5 2 1 4 | 3 |