

Задача А. Подсчет хештегов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дима разрабатывает новую модную соцсеть. В данный момент он хочет написать модуль, который будет обрабатывать текст и подсчитывать количество различных хештегов в нем.

Текст состоит из строчных английских букв, цифр, символов «#» и пробелов. Назовем словом несколько последовательных непробельных символов, ограниченных слева и справа пробелами или концом/началом текста. Слово является хештегом, если его длина хотя бы 2, первый символ равен «#», а все остальные — нет.

Дима хочет вычислить количество различных хештегов, которые встречаются в тексте, вывести их, а также количество раз, которое встречается каждый из них.

Формат входных данных

В единственной строке дан текст в формате, описанном выше. Длина текста не превышает 100 000. Текст может содержать несколько пробелов подряд, пробелы в начале или конце строки.

Формат выходных данных

В первой строке выведите целое число n — количество различных хештегов, которые встречаются в тексте. В следующих n строках выведите хештег и количество раз, которое он встречается. Хештеги можно выводить в любом порядке.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	50	Длина текста не превышает 1 000		первая ошибка
2	50	Без дополнительных ограничений	1	первая ошибка

Примеры

стандартный ввод	
i have brought #peace #freedom #justice and #security to my new empire	
стандартный вывод	
4 #freedom 1 #justice 1 #peace 1 #security 1	
стандартный ввод	
#to #be #or #not #to #be	
стандартный вывод	
4 #be 2 #not 1 #or 1 #to 2	

стандартный ввод
#ioip# #2021 #03#28
стандартный вывод
1 #2021 1

Задача В. Раскладывание приборов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Олимпиада закончилась, и, пока жюри подводит итоги, участники решили сходить в ближайший ресторан.

Из-за эпидемиологических ограничений, в ресторане есть только один круглый стол на n мест и ровно два официанта. Места за столом пронумерованы натуральными числами от 1 до n , в порядке обхода по часовой стрелке. Таким образом, соседними являются места i и $i + 1$ для всех $1 \leq i < n$, а также места n и 1.

Управляющий рестораном, увидев, что к ним пришло так много посетителей, поручил официантам разложить перед каждым местом столовые приборы. Изначально приборов нет нигде, первый официант стоит около места номер a , а второй около места номер b . За секунду официант может перейти от одного места к соседнему слева или справа. У каждого официанта есть тележка с бесконечным количеством приборов, и, если он стоит около места, на котором еще нет приборов, он их мгновенно выкладывает на это место.

Пока участники ждут, им стало интересно, какое минимальное количество секунд необходимо официантам, чтобы разложить перед каждым местом столовые приборы. Помогите им найти ответ на этот вопрос.

Формат входных данных

В первой строке ввода дано ровно одно целое число n — количество мест за круглым столом ($1 \leq n \leq 10^{18}$).

Во второй строке через пробел записаны два целых числа a и b — номера мест, рядом с которыми изначально находятся первый и второй официанты ($1 \leq a, b \leq n$).

Формат выходных данных

Выведите единственное целое число — минимальное количество секунд, которое необходимо официантам, чтобы разложить перед каждым местом столовые приборы.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	9	$n \leq 10$		первая ошибка
2	14	Места a и b совпадают или являются соседними		первая ошибка
3	17	$n \leq 1\,000$	1	первая ошибка
4	31	$n \leq 100\,000$	1, 3	первая ошибка
5	29	Без дополнительных ограничений	1–4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 1 3	2
7 3 1	3

Замечание

Один из способов разложить приборы в первом примере:

- Маршрут первого официанта: $1 \rightarrow 2 \rightarrow 3$.
- Маршрут второго официанта: $3 \rightarrow 4 \rightarrow 5$.

Один из способов разложить приборы во втором примере:

- Маршрут первого официанта: $3 \rightarrow 4 \rightarrow 5 \rightarrow 6$.
- Маршрут второго официанта: $1 \rightarrow 2 \rightarrow 1 \rightarrow 7$.

Задача С. Шестизначные документы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Бухгалтер Валерий разбирается с нестыковками в бухгалтерских отчетах. Ему осталось проверить ровно n документов, i -й из которых доступен в корпоративной сети по шестизначному целочисленному идентификатору a_i .

Назовем *инверсией* в k -м разряде пару номеров i и j такую, что $i < j$, и k -я цифра числа a_i строго больше k -й цифры числа a_j . Тогда *сложностью* массива шестизначных чисел $\{a_i\}$ назовем суммарное количество инверсий во всех шести разрядах.

Валерий знает, что чем меньше сложность набора идентификаторов, тем меньше времени он потратит на вбивание их в адресную строку. Поскольку множество документов фиксировано, а радикально менять порядок проверки опасно (можно случайно пропустить некоторые документы), единственный доступный Валерию способ изменить исходный порядок проверки — сдвинуть его по циклу на несколько позиций. Напомним, что *циклическим сдвигом* массива a_1, a_2, \dots, a_n на t позиций влево называется массив $a_{t+1}, a_{t+2}, \dots, a_n, a_1, a_2, \dots, a_t$.

Помогите Валерию выбрать циклический сдвиг исходного массива идентификаторов с минимальной сложностью.

Формат входных данных

В первой строке ввода дано целое число n — количество документов, которые требуется проверить ($1 \leq n \leq 100\,000$).

В i -й из следующих n строк даны шесть цифр — идентификатор a_i . Гарантируется, что все a_i различны. Идентификаторы могут начинаться с нуля.

Формат выходных данных

Выведите единственное целое число — минимальную из сложностей циклических сдвигов массива идентификаторов документов.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	17	$n \leq 50$		первая ошибка
2	19	$n \leq 300$	1	первая ошибка
3	26	$n \leq 1\,000$	1, 2	первая ошибка
4	38	Без дополнительных ограничений	1–3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 277659 177013 314836	4
3 250401 185217 296632	3

Замечание

В первом примере выгодно сделать сдвиг на одну позицию влево, тогда число 177013 окажется на первом месте. В таком случае в первых четырех разрядах будет по одной инверсии, а в последних двух — ноль.

Во втором примере порядок чисел уже оптимален с тремя инверсиями: по одной в первом, четвертом и шестом разрядах.

Задача D. Государственный переполох

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В стране Байтландии есть ровно n городов. Страной управляет президент, а в каждом городе сидит некоторое (не обязательно одинаковое, возможно нулевое) количество министров, которое мы назовем *размером министерства* этого города. В любом отдельно взятом городе министры пронумерованы подряд целыми положительными числами, начиная с 1. В подчинении у министра находятся все министры его города с меньшими номерами. Известно, что ни в каком городе не находится более h министров.

Назовем *важностью* министра количество его подчиненных, тогда *важность* i -го министра в любом городе в точности равна $i - 1$. Чтобы члены правительства не перекладывали обязанности друг на друга, были введены следующие правила:

1. Ни один министр не знает своей важности.
2. У каждого министра есть способ связи со всеми министрами такой же важности из других городов, и только с ними.
3. У президента в каждый момент времени есть способ связи с министрами максимальной важности из каждого города, и только с ними.

Сегодня президент решил выяснить сколько всего в Байтландии министров. Для этого он может делать два типа запросов:

1. «- i x » — уменьшить размер министерства i -го города на x . В этом случае, если x больше количества министров в городе, то ничего не происходит, иначе — в отставку уходят x министров **с максимальной важностью**. Разумеется, после этого президент получает контакт нового самого важного министра города i , а министры из других городов теряют связь с ушедшими в отставку.
2. «? i » — спросить у самого важного министра города i , с каким числом министров (включая его самого) у него есть связь. Если же в городе i не осталось министров, автоответчик сообщит президенту число n . Таким образом, в любом случае, президент узнает в ответ количество городов, в которых **на данный момент** есть хотя бы столько же министров, сколько и в i -м, включая и сам i -й город.

Помогите президенту определить, сколько в Байтландии министров. Даже если их количество уменьшится в процессе выяснения ответа, президент хочет знать, сколько их было изначально. Разумеется, время президента очень ценно, поэтому он успеет сделать не более q описанных выше запросов.

Формат входных данных

Это интерактивная задача. Помимо этого, каждый тест состоит из нескольких наборов данных.

В первой строке ввода дано целое число t — количество наборов данных в тесте ($1 \leq t \leq 5$). Гарантируется, что во всех тестах, кроме примера, $t = 5$.

Во второй строке ввода через пробел даны целые числа n , h и q — количество городов, ограничение на количество министров в городе и ограничение на число запросов ($1 \leq n \leq 5000$; $1 \leq h \leq 1\,000\,000$; $1 \leq q \leq 24\,000$). Эти ограничения являются общими для всех наборов данных текущего теста.

Далее t раз запускается протокол взаимодействия с интерактором.

Формат выходных данных

Когда ваша программа готова дать ответ на задачу, следует вывести «! a » (без кавычек), где a — предполагаемый ответ. После этого программа должна перейти к следующему набору данных или завершиться в соответствии с описанными в следующей секции правилами.

Протокол взаимодействия

Интерактор ожидает от вашей программы запросы трех типов: «- i x », «? i » и «! a », где i , x и a — целые числа ($1 \leq i \leq n$; $0 \leq x \leq h$). После каждого запроса должен следовать перевод строки. При несоблюдении вашей программой формата запросов, ваше решение может получить произвольный вердикт (отличный от ОК).

На запрос первого типа интерактор на новой строке отвечает «ОК», если операция прошла успешно, и «FAIL», если в i -м городе было меньше x министров. На запрос второго типа интерактор также на новой строке ответит числом, равным количеству городов, в которых не меньше министров, чем в i -м городе.

Запрос третьего типа означает, что ваша программа готова дать ответ на задачу. Если a — верный ответ, интерактор выведет «ОК» и перейдет к следующему тесту данного мультитеста или завершится, если такого нет. Если ответ неверен, интерактор выводит «-1» и завершается с вердиктом WA.

Ваша программа может вывести не более q запросов первого или второго типа. Запрос третьего типа (ответ на тест) в этом ограничении не учитывается. При превышении данного ограничения, интерактор выводит «-1» завершается с вердиктом WA.

Обратите внимание, что завершение интерактора означает, что следующие тесты данного мультитеста будут пропущены. Чтобы не получить при этом вердикт TL или PL, при прочтении из ввода значения «-1» ваша программа должна завершить свою работу с нулевым кодом возврата.

Система оценки

В этой задаче 25 тестов, помимо теста из примера. Каждый тест оценивается независимо по указанным ниже критериям и оценивается максимум в 4 балла. Ограничения на входные данные, используемые в каждом тесте, указаны в таблице в конце этой секции.

Баллы за тест начисляются только если на каждом из t наборов данных был дан верный ответ без превышения лимита запросов. В таком случае количество баллов за каждый набор данных вычисляется по формуле

$$\text{score}(c, j) = \max \left(1, 4 - \left\lfloor \max \left(0, \frac{c - j}{\gamma} \right) \right\rfloor \right)$$

где c — количество итераций, сделанное вашей программой, j — количество итераций, сделанное решением жюри, а γ — отдельно заданный для каждого теста параметр.

В качестве финальной оценки за тест берется минимум из t оценок его наборов данных.

Номер теста	n	h	q	γ	Дополнительные ограничения
2	10	10	24 000	12 000	-
3	20	1000	20 000	6000	-
4	1000	20	20 000	4000	-
5	1000	100	6 000	2000	Размеры министерств в разных городах принимают не более 50 различных значений
6	5000	500	6000	5	-
7	5000	5000	15 000	100	-
8	1800	20 000	24 000	250	Размеры министерств любых двух разных городов отличаются хотя бы на 5
9	100	1000	1000	300	-
10	500	1 000 000	10 000	2500	-
11	300	800 000	4000	600	Среднее геометрическое размеров министерств не превосходит 20
12	350	1 000 000	5000	700	Среднее геометрическое размеров министерств не превосходит 35
13	100	200	380	1	-
14	100	5000	700	10	-
15	2000	500 000	10 000	100	-
16	100	10 000	1000	50	-
17	2000	500 000	10 500	150	-
18	2000	1 000 000	10 030	50	-
19	2000	1 000 000	10 030	20	-
20	100	1 000 000	1820	5	-
21	200	1 000 000	3420	15	-
22	200	1 000 000	3420	5	-
23	50	1 000 000	1000	30	-
24	360	1 000 000	6500	20	-
25	720	1 000 000	12 000	30	-
26	900	1 000 000	14 000	40	-

Пример

стандартный ввод	стандартный вывод
1	
3 3 6	
	? 1
3	
	- 1 1
OK	
	- 2 3
OK	
	? 1
3	
	- 3 3
FAIL	
	- 3 2
OK	
	! 6
OK	

Замечание

Это интерактивная задача. При использовании буферизованного вывода не забывайте сбрасывать буфер при выводе запросов (`sys.stdout.flush()` в Python, `System.out().flush()` в Java и `std::cout.flush()` в C++).

В примере из условия происходят следующие действия:

1. Первым запросом выясняется, что первое министерство — самое маленькое, так как во всех трех городах хотя бы столько же министров.
2. Следующими двумя действиями в отставку отправляется один министр из первого города и три из второго. Из того, что $h = 3$, и ответ интерактора на запрос «- 2 3» — это «OK», можно сделать вывод, что во втором городе было **ровно 3** министра.
3. После этого выясняется, что во всех городах не меньше министров, чем в первом городе. Но мы знаем, что во втором их теперь 0, а значит и в первом стало 0, то есть было 1.
4. Последними двумя запросами после неудачной попытки отправить трех министров из третьего города, и удачной — двух, мы понимаем, что их было ровно 2.

Таким образом, ответ на тест из условия — $1 + 3 + 2 = 6$. Обратите внимание также, что $q = 6$, и было сделано ровно 6 запросов вида «-» и «?», тогда как запрос «!» в это количество не входит.

Задача Е. Загадочное устройство

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

У Димы есть устройство, содержащее $n + m$ кнопок и экран. На экране отображается строка s , изначально она пустая. Если нажать на кнопку с номером i ($1 \leq i \leq n$), то в конец строки s дописется строка w_i . Если нажать на кнопку с номером $n + i$ ($1 \leq i \leq m$), то из конца строки s удалится a_i символов. Если в строке s было меньше a_i символов, она станет пустой.

У Димы есть q любимых строк t_i . Для каждой из них он хочет узнать, можно ли в результате нескольких нажатий на кнопки из пустой строки s сделать строку равную t_i .

Формат входных данных

В первой строке дано два целых числа n и m — количество кнопок первого и второго типа ($1 \leq n, m \leq 100\,000$).

В следующих n строках даны непустые строки w_i , состоящие из строчных английских букв. Сумма длин всех строк w_i не превышает 10^6 .

В следующих m строках даны числа a_i ($1 \leq a_i \leq 10^6$).

В следующей строке дано целое число q — количество любимых строк Димы ($1 \leq q \leq 100\,000$).

В следующих q строках даны непустые строки t_i , состоящие из строчных английских букв. Сумма длин всех строк t_i не превышает 10^6 .

Формат выходных данных

Для каждой строки t_i в новой строке выведите «Yes», если можно s сделать равным t_i , и «No» иначе.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	12	$\sum w_i \leq 1\,000, \sum t_i \leq 1\,000,$ $m = 1, w_i = a_1$		первая ошибка
2	13	$\sum w_i \leq 1\,000, \sum t_i \leq 1\,000,$ $m = 1, a_1 = 1$		первая ошибка
3	19	$\sum w_i \leq 1\,000, \sum t_i \leq 1\,000$	1	первая ошибка
4	29	$\sum w_i \leq 100\,000, \sum t_i \leq 100\,000$	1, 2	первая ошибка
5	27	Без дополнительных ограничений	1–3	первая ошибка

Пример

стандартный ввод	стандартный вывод
2 1	Yes
bba	Yes
abacaba	Yes
2	No
4	
b	
ababbab	
abacabba	
caba	