

Задача А. Поход в кино

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

Сегодня премьера нового фильма «Тазики», и Молния Маккуин обязательно хочет на него сходить. Однако, одному идти скучно, поэтому он решил взять с собой друзей. Все бы хорошо, но Маккуин знает, что на премьеру обязательно пойдет его давний враг Чико Хикс, с которым он и его друзья совсем не хотят видеться, а тем более сидеть рядом.

Маккуин хорошо знает Чико, поэтому уверен, что тот сядет в центр зала. Так же он знает, что в зале будет ровно N рядов по N мест. Хорошенько все обсудив с друзьями, Маккуин решил, что они все-таки пойдут на премьеру. Но с одним условием. Он и его друзья будут сидеть хотя бы в A рядах и хотя бы в B местах от Чико. Это означает, что если Маккуин или его друг сидит на ряду номер x_1 , на месте номер y_1 , а Чико сидит на ряду номер x_2 , месте номер y_2 , то должны выполняться два условия: $|x_1 - x_2| \geq A$, $|y_1 - y_2| \geq B$.

Теперь Маккуин хочет понять, какое максимальное количество друзей он может взять с собой.

Формат входного файла

В первой и единственной строке входного файла дано три числа N, A, B ($1 \leq N, A, B \leq 1000$, N — нечетное) — количество рядов и мест в зале, а также ограничения из условия.

Формат выходного файла

В единственной строке выходного файла выведите количество друзей, которое Маккуин сможет взять с собой (не считая его самого). Если подходящего места нет даже для Маккуина, в единственной строке выходного файла выведите 0.

Пример

<code>cinema.in</code>	<code>cinema.out</code>
3 1 1	3
3 1 2	0

Задача В. Оцепление

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

Из тюрьмы сбежала особо опасная тачка — Маттэо и уже мчится по Шоссе 66 к городку Радиатор-Спрингс. Полицейский Шериф хочет поймать Маттэо, когда тот будет проезжать по городу.

Радиатор-Спрингс разбит на $n \times m$ кварталов, то есть план города представляет собой матрицу $n \times m$. Квартал с координатами $(1, 1)$, который находится в левом верхнем углу — это въезд в город, квартал с координатами (n, m) , который находится в правом нижнем углу — выезд из города.

Маттэо въедет в город и начнет свое движение к выезду. Он очень спешит, поэтому будет двигаться только вправо или вниз, то есть находясь в квартале с координатами (i, j) , он поедет в квартал $(i + 1, j)$ или $(i, j + 1)$. Но Шериф пока не знает, какой конкретно маршрут выберет Маттэо, поэтому Шериф хочет оцепить некоторые кварталы. Он хочет сделать это так, чтобы какой бы маршрут ни выбрал Маттэо, он обязательно проедет как минимум по k оцепленным кварталам. При этом Шериф хочет минимизировать количество оцепленных кварталов, чтобы как можно меньше будоражить город. Так же он не собирается оцеплять въезд и выезд, чтобы не затруднять движение других тачек.

Помогите Шерифу поймать Маттэо.

Формат входного файла

Единственная строка входного файла содержит три натуральных числа n, m, k ($1 \leq n, m \leq 300$, $0 \leq k \leq 10^9$, $n \times m > 1$) — размер города и минимальное количество оцепленных кварталов, которое должно встретиться Маттэо на пути от въезда до выезда.

Формат выходного файла

В первой строке выведите «YES», если решение существует или «NO» в противном случае. В случае положительного ответа в следующих строках выведите матрицу $n \times m$ — план оцепления, каждый символ которой либо 'C', обозначающий оцепленный квартал, либо '.' — свободный квартал.

Пример

<code>cordon.in</code>	<code>cordon.out</code>
2 2 1	YES .C C.
1 6 2	YES ..CC..
3 3 4	NO

Задача С. Экзамен в Британской разведке

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

Финн Максисл — агент Британской разведки. Любой агент разведки обязан быть готов к самым непредвиденным ситуациям и уметь в считанные секунды решить любую поставленную задачу. Кроме чисто физических нормативов, на выпускном экзамене разведывательной академии требуют умения решать и сложные алгоритмические задания — мало ли, что пригодится в жизни? Финну досталась следующая:

Дана строка s длины n . Назовем ее k -разбиваемой, если ее можно разбить на k непрерывных подстрок равной длины так, что любую подстроку из любой другой можно получить циклическим сдвигом. Нужно найти все k , что строка s — k -разбиваемая.

Поскольку в разведке все методы хороши, агент Финн решил попросить помощи у вас.

Строка s называется циклическим сдвигом строки t , если существуют такие (возможно, пустые) строки u, v , что $t = uv$ и $s = vu$.

Формат входного файла

В первой строке дана строка s ($1 \leq |s| \leq 200\,000$), состоящая из строчных латинских букв.

Формат выходного файла

В первой строке выведите число m — количество подходящих чисел k .

Во второй строке выведите через пробел m подходящих чисел k_i , отсортированных по возрастанию.

Пример

<code>kdivision.in</code>	<code>kdivision.out</code>
abbabaab	3 1 2 4
abbababa	2 1 4

Комментарий

В первом примере можно оставить строку как есть, разбить на две строки «abba» и «baab» (несложно убедиться, что они могут быть получены друг из друга циклическим сдвигом), или разбить на четыре строки «ab», «ba», «ba» и «ab», которые так же могут быть получены друг из друга циклическим сдвигом.

Задача D. Долгое путешествие

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

Мэтр очень любит путешествовать и смотреть на мир вокруг. Но еще больше он любит смотреть на изменение своего счетчика пробега. Недавно он заметил одно интересное свойство у последних чисел, которые он видел на счетчике. Оказалось, что в каждом из них количество различных цифр не превосходило k . Мэтр решил, что это все неспроста, и назвал такие числа «замечательными». После этого он посмотрел на текущее показание счетчика и задался вопросом: а когда в следующий раз на нем будет показано замечательное число? Понимая, что эта задача ему не под силу, Мэтр обратился к вам за помощью.

Формат входного файла

В первой строке входного файла дано число k ($1 \leq k \leq 10$) — ограничение на количество различных цифр в замечательных числах.

Во второй строке дано число x ($1 \leq x \leq 10^{10^6}$) — последнее показание счетчика, которое увидел Мэтр. Гарантируется, что в числе x , нет ведущих нулей.

Формат выходного файла

В единственной строке выходного файла выведите единственное число y ($y \geq x$) — показание счетчика, такое, что в числе y не более k различных цифр и $(y - x)$ — минимально. Ответ не должен содержать ведущих нулей.

Пример

<code>knumbers.in</code>	<code>knumbers.out</code>
1 4321	4444
2 1234	1311
3 123	123

Задача Е. Расчеты Луиджи

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

Луиджи знает толк в подборе шин. Недавно он попросил Гвидо проводить расчеты для определения оптимального сочетания трех факторов: дорога, шины и диски.

В расчетах Луиджи дорога характеризуется числом k — частотой микровыбоин, а шины и диски числами x и y — коэффициентами крутости по личной шкале Луиджи. Гвидо опытно установил, что обязательным условием сочетания факторов является выполнение формулы: $x + y \equiv 0 \pmod k$. То есть сумма коэффициентов крутости шин и дисков должна нацело делиться на частоту микровыбоин.

Луиджи всегда работает на максимуме возможностей и хочет подбирать в своем магазине самые крутые шины, учитывая ассортимент. В магазине Луиджи можно купить шины с любыми коэффициентами крутости от A до B , аналогично диски коэффициентами от C до D . При этом если вариантов, подходящих под данную дорогу несколько, то Луиджи подбирает любой из тех, в котором сумма коэффициентов максимальна.

Луиджи просит вас, написать программу, которая по числам A, B, C, D и k определит какие шины и какие диски нужно продать.

Формат входного файла

В первой и единственной строке входного файла дано пять натуральных чисел A, B, C, D, k ($1 \leq A, B, C, D, k \leq 10^9$) — ограничения на ассортимент, а также частота микровыбоин дороги.

Формат выходного файла

В единственной строке выходного файла выведите два числа x и y ($A \leq x \leq B, C \leq y \leq D$) — ответ на задачу. Если ответа не существует, выведите -1 .

Пример

<code>luigi.in</code>	<code>luigi.out</code>
1 10 1 7 3	9 6
1 2 2 3 7	-1
1 4 2 3 2	4 2

Задача F. Железная дорога

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

Через Радиатор-Спрингс проходит крупнейшая железная дорога. Введем на ней систему координат. По железной дороге движутся n поездов. Каждый поезд представляет собой отрезок. Поезд с номером i в начальный момент времени занимает отрезок $[a_i; b_i]$. Поезда не стоят на месте — i -й поезд движется с постоянной скоростью v_i . Железная дорога двунаправлена, то есть поезда могут двигаться как в положительном направлении оси, так и в отрицательном. Отрезки, представляющие поезда, в любой момент времени могут пересекаться, вкладываться друг в друга и совпадать.

В точке x находится железнодорожный переезд, к которому в моменты t_i подъезжают тачки. Для каждой тачки требуется вычислить минимальный момент времени, в который она сможет пересечь железнодорожный переезд.

Тачка может пересечь переезд, если он не занят поездом. Переезд считается занятым, если отрезок, представляющий собой некоторый поезд, содержит в себе точку x . Причем если поезд подъезжает к перекрестку одновременно тачкой, то переезд считается занятым. Тачки пересекают переезд мгновенно.

Формат входного файла

В первой строке входного файла три целых числа — n , m и x ($1 \leq n, m \leq 10^5, |x| \leq 10^9$) — количество поездов, движущихся по железной дороге, количество тачек, подъезжающих к железнодорожному переезду и точка, в которой находится переезд.

В следующих n строках содержатся по три целых числа a_i, b_i, v_i ($|a_i| \leq 10^9, |b_i| \leq 10^9, 1 \leq v_i \leq 10^9, a_i \neq b_i$) — отрезок, задающий поезд и его скорость движения. Если $a_i < b_i$, то поезд движется в положительном направлении оси, если $a_i > b_i$ — в отрицательном.

В следующей строке находятся m неотрицательных целых чисел t_j ($0 \leq t_j \leq 10^9$) — моменты времени, в которые к переезду подъедут тачки.

Формат выходного файла

В m строках выходного файла выведите m вещественных чисел b_j — минимальный момент времени, в который j -я тачка сможет пересечь железнодорожный переезд. Ответ будет считаться правильным, если относительная или абсолютная погрешность каждого b_j не превосходит 10^{-6} .

Пример

<code>railroad.in</code>	<code>railroad.out</code>
3 2 0 -4 -1 1 13 6 3 -7 -6 1 1 5	4.333333333 5.000000000
2 2 0 4 2 1 -11 -8 2 2 6	5.500000000 6.000000000

Задача G. Детали

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

После победы в очередной гонке, Молния Маккуин решил обновить покрышки и другие свои детали. Придя в магазин, Молния был обескуражен количеством различных товаров. Радостный, он бросился набирать себе в багажник всякие железные штучки. Но в один момент что-то начало тревожить гоночную машину. Это чувство тревоги не покидало Молнию пока, стоя в очереди в кассу, он не осознал, что забыл все свои гаечки дома. А без гаечек невозможно совершить покупку, ведь это главная валюта в Карбюраторном округе! Но Молния Маккуин решил не унывать. Он запомнил все цены на детали и поехал домой за гаечками.

Естественно, на момент прибытия домой, Молния почти все забыл. Единственное, что он помнил про каждую деталь — диапазон цен, которому принадлежит истинная цена. Другими словами, про деталь с номером i Молния знает, что ее стоимость не меньше чем a_i и не больше чем b_i гаечек. Гаечки в Карбюраторном округе бывают номиналом в 2^k условные единицы, где $k \geq 0$.

Маккуину стало интересно, какое минимальное количество гаечек нужно взять, чтобы можно было расплатиться за всю покупку без сдачи при любых стоимостях из указанных диапазонов. Поскольку Молния Маккуин очень успешный гонщик, можно считать, что гаечек каждого номинала у него неограниченно много.

Формат входного файла

В первой строке входных данных дано число n ($1 \leq n \leq 10^5$) — количество покупок. Далее следует n строк. В $i + 1$ -й строке даны числа a_i, b_i ($1 \leq a_i \leq b_i \leq 10^9$) — диапазон стоимостей i -й детали.

Формат выходного файла

Выведите минимальное число гаечек, которое нужно взять, чтобы расплатиться за всю покупку без сдачи при любых стоимостях из указанных диапазонов.

Пример

<code>parts.in</code>	<code>parts.out</code>
4	5
3 5	
7 9	
2 4	
1 10	

Задача Н. Гонка

Имя входного файла:	<code>race.in</code>
Имя выходного файла:	<code>race.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Радиатор-Спрингс назревает гонка. Каждая уважающая себя тачка в городе хотела бы поучаствовать в ней. На данный момент на нее записались n тачек. У каждой тачки есть автомобильный номер, в Радиатор-Спрингс это непустая строка, состоящая из строчных латинских букв. Номера двух тачек необязательно различны.

Но участие в гонке смогут принять не все. Судьи мероприятия сами выбирают машины, которые примут в ней участие. Они хотят максимизировать *зрелищность* гонки. *Зрелищность* гонки — это натуральное число, которое равно произведению трех величин:

- количество машин, участвующих в гонке;
- длина наибольшего общего префикса номеров машин, участвующих в гонке;
- длина наибольшего общего суффикса номеров машин, участвующих в гонке.

Наибольшим общим префиксом множества строк $s_1, s_2 \dots s_n$ называется наибольшая по длине строка, которая является префиксом каждой строки s_i .

Наибольшим общим суффиксом множества строк $s_1, s_2 \dots s_n$ называется наибольшая по длине строка, которая является суффиксом каждой строки s_i .

Такое определение *зрелищности* связано с фотографиями, которые делаются во время гонок. Чем больше «похожи» номера мчащихся рядом тачек, тем больше эстетического удовольствия доставляют фотографии.

Помогите судьям выбрать подмножество машин с наибольшей *зрелищностью*.

Формат входного файла

В первой строке входного файла содержится одно натуральное число n ($1 \leq n \leq 2 \cdot 10^5$) — количество машин, желающих принять участие в соревновании. В следующих n строках содержатся n непустых строк, состоящих из строчных латинских букв.

Суммарная длина строк не превосходит $2 \cdot 10^5$.

Формат выходного файла

Единственная строка выходного файла должна содержать целое число — наибольшую *зрелищность* гонки, которую можно достичь.

Пример

race.in	race.out
4 aa aa aa bb	12
4 aba ab abbba aba	25
8 abcccbx abccbх abcycbx abczcbx abcbx abcacbx abcscbx axcccbx	63

Комментарий

В первом примере в гонке будут участвовать первые три машины. Длина их наибольшего общего префикса — 2, суффикса — 2, количество — 3, получаем $3 \times 2 \times 2 = 12$.

Во втором примере в гонке будет участвовать только третий автомобиль. Наибольший общий префикс и суффикс одной строки совпадает с этой строкой, поэтому получаем $1 \times 5 \times 5 = 25$.

В третьем примере в гонке будут участвовать все автомобили, кроме последнего. Длина их наибольшего общего префикса — 3, суффикса — 3, всего автомобилей — 7, получаем $7 \times 3 \times 3 = 63$.

Задача I. Новобранцы

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

По заказу правительства США был разработан новый инновационный вид тачек. Для того, чтобы натренировать и испытать их в действии, эти тачки были направлены главной военной тачке страны — Сержанту. Узнав об этом, Сержант очень обрадовался. Однако очень быстро от его радости не осталось и следа. Больше всего Сержанта расстроила недисциплинированность новобранцев. Им было принято решение это исправить.

Сержант построил n тачек в шеренгу. Из них m автомобилей являются новобранцами. Соответственно, $(n - m)$ машин «старой закалки». Сержант отдал k команд, каждая из которых имеет вид:

- «Налево!» — поворот на 90° налево;
- «Направо!» — поворот на 90° направо;
- «Кругом!» — поворот на 180° .

Известно, что старые машины выполняют все команды правильно, а новые всё делают наоборот. Иными словами, когда новая машина слышит команду «Налево!», она поворачивает направо, когда слышит команду «Направо!», она поворачивает налево. Команду «Кругом» новые машины выполняют правильно. Этот беспредел Сержанту очень не понравился. После каждой команды он смотрел на всю шеренгу, и если в ней было две машины, направленные в разные стороны, он заставлял всю шеренгу как следует хлопнуть дверьми. Сержанту стало интересно, сколько раз ему пришлось наказывать своих подчинённых. Помогите ему посчитать это количество, чтобы бедным машинам не пришлось снова терпеть наказания!

Формат входного файла

В первой строке входных данных даны числа n, m ($1 \leq n \leq 10^5, 0 \leq m \leq n$) — общее количество машин в шеренге и количество новобранцев. Во второй строке дано число k ($1 \leq k \leq 10^5$) — число команд, отданных Сержантом. В третьей строке дано описание команд — строка длиной ровно k символов, каждый символ в которой обозначает команду: «L» — налево, «R» — направо, «A» — кругом. Гарантируется, что строка состоит только из данных символов.

Формат выходного файла

Выведите количество раз, которое Сержанту пришлось наказывать своих подчинённых.

Пример

<code>soldiers.in</code>	<code>soldiers.out</code>
5 3	3
7	
LRARLRL	