

[C] 1. Петя и кубики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Маленький Петя одолжил в детском саду у своего друга Егора набор кубиков с цифрами. Набор состоит из длинной коробки и 10 кубиков, которые размещаются в ней в один ряд.

К сожалению, с Петей всегда происходят какие-то неприятности, вот и в этот раз в процессе игры он потерял один кубик. Вечером, собирая кубики обратно в коробку, Петя обнаружил пропажу, потому что одно место в коробке осталось пустым. Так как Петя ещё маленький и не знает цифр, он не знает, какой кубик был потерян. Он помнит только, что все цифры на них были разные. Петя очень не хочет, чтобы Егор расстраивался, поэтому он просит вас помочь ему понять, какой кубик был потерян.

Формат входных данных

Дана строка S длины 9, состоящая из цифр, написанных на Петиних кубиках. Гарантируется, что все цифры различные.

Формат выходных данных

Выведите одно число – цифру на недостающем кубике.

Примеры

стандартный ввод	стандартный вывод
123456789	0
098765432	1

[C] 2. Гламур

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Одно известное модное агентство решило закупить очередную партию платьев для своих n моделей. Чтобы ни у кого из моделей не возникло претензий, агентство решило, что каждой модели должно достаться поровну платьев.

Кроме того, закупать платья у поставщика можно только парами, ведь одно и то же платье выпускается в версии для блонд и для брюнеток. При этом неважно, какая модель какие платья получит, потому что все модели в агентстве — рыжие.

Определите минимальное количество платьев, которое нужно купить агентству.

Формат входных данных

В единственной строке входных данных задано одно целое число n ($1 \leq n \leq 10^9$) — количество моделей в агентстве.

Формат выходных данных

Выведите одно целое число — минимальное число платьев, которое нужно закупить агентству.

Примеры

стандартный ввод	стандартный вывод
3	6
10	10

Замечание

В первом примере входных данных можно купить 3 пары платьев, тогда каждой модели достанется по 2 платья. Можно показать, что меньше платьев купить нельзя.

[C] 3. Забыли пароль?

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вася Пупкин, как обычно, забыл пароль от сайта — на этот раз от своей любимой социальной сети ТокТёк. Убедившись, что дефолтная комбинация не подходит, Василий решил напрячь все умственные силы и всё-таки вспомнить пароль.

У Василия крайне нестандартное ассоциативное мышление, поэтому пароль он не вспомнил, зато вспомнил некоторую информацию про цифры, содержащиеся в пароле. Он записал это в виде строки $s_0s_1 \dots s_9$, где $s_i = o$, если Вася уверен, что цифра i содержится в пароле хотя бы один раз, $s_i = x$, если Вася уверен, что цифра i не содержится в пароле, и $s_i = ?$, если Вася точно не помнит, содержится i -я цифра в пароле или нет.

Держа в уме, что пароли Василия всегда состоят ровно из 4-х цифр, подскажите, сколько паролей подходят под его воспоминания.

Формат входных данных

В единственной строке входных данных содержится строка s , состоящая ровно из 10 символов, каждый из которых либо o , либо x , либо $?$

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
?x?x?ooxox	108
?oxooooo??	0

Замечание

В первом примере входных данных, подойдут, например, пароли 6852 или 5086.

Во втором примере входных данных не подходит ни одного пароля (память подвела Василия).

[C] 4. Hogwarts Legacy 2

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Только что вышло продолжение игры «Hogwarts Legacy». В ней, играя за главного героя, вы должны бороться с монстрами и злыми волшебниками. Во второй версии добавили новый механизм ограничения ресурсов: у игрока может быть несколько единиц энергии, которые он может тратить на заклинания и затем восполнять. Изначально у игрока 0 единиц энергии.

Любое заклинание стоит 1 единицу энергии. Ограничения сверху на число единиц энергии у игрока нет, но оно не может быть меньше нуля. Чтобы пополнять единицы энергии, надо пить сливочное пиво. Каждая кружка сливочного пива добавляет фиксированное число единиц энергии b .

Вам дан журнал игры с действиями четырех видов:

- **cast** – наложить заклинание. Гарантируется, что на этот момент есть хотя бы одна единица энергии
- **drink** – выпить кружку целебного сливочного пива
- **save** – сохранить игру
- **load** – загрузить последнее сохранение. Если до этого не было ни одного сохранения, то игра откатывается на начальное состояние.

После каждого действия выведите, сколько единиц энергии осталось у игрока.

Формат входных данных

В первой строке даны число действий в журнале – n ($1 \leq n \leq 100$), и сколько единиц энергии прибавляет каждая кружка сливочного пива – b ($1 \leq b \leq 100$).

В следующих n строках описаны сами действия (каждое действие – одно из четырех допустимых).

Формат выходных данных

Выведите n строк – для i -го действия выведите, сколько единиц энергии стало у героя после него.

Пример

стандартный ввод	стандартный вывод
8 4	4
drink	4
save	3
cast	2
cast	6
drink	6
save	5
cast	6
load	

Замечание

В первом примере будет происходить следующее:

- Изначально было 0 единиц энергии.
- После действия 1: игрок выпил сливочное пиво и увеличил число единиц энергии на 4. Стало 4

- После действия 2: игрок сохранился, число единиц энергии осталось прежним, то есть 4
- После действия 3: игрок наложил заклинание и потратил единицу энергии, осталось 3
- После действия 4: игрок наложил заклинание и потратил единицу энергии, осталось 2
- После действия 5: игрок выпил сливочное пиво и увеличил число единиц энергии на 4. Стало 6
- После действия 6: игрок сохранился, число единиц энергии осталось прежним, то есть 6
- После действия 7: игрок наложил заклинание и потратил единицу энергии, осталось 5
- После действия 8: игрок загрузил последнее сохранение (которое было на 6-м действии), у него стало 6 единиц энергии

[C] 5. Кубы палиндромы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано число N . Найдите максимальное число x , такое что выполняются следующие условия:

- $x \leq N$
- $x = y^3$ для некоторого целого y (является кубом некоторого числа)
- x является палиндромом, то есть читается одинаково слева направо и справа налево (например, числа 404, 1001 – являются палиндромами, а 100, 443 – нет)

Формат входных данных

В единственной строке дано число N ($1 \leq N \leq 10^{18}$)

Формат выходных данных

В единственной строке выведите одно число – ответ на задачу

Примеры

стандартный ввод	стандартный вывод
433	343
83293823928323	1334996994331

[C] 6. Скрытый смысл

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Шестиклассница Лиза любит искать везде скрытый смысл. Недавно на кружке математики она научилась решать числовые ребусы.

Числовой ребус – это задача, условие которой записывается, как сложение двух чисел в столбик, но вместо чисел записаны слова. Чтобы решить ребус, нужно заменить каждую букву ребуса на цифру, так чтобы в итоге получилось верное равенство. (При этом каждая цифра может соответствовать только одной букве.) Ребус может иметь одно или несколько подходящих решений, а также не иметь решений вовсе.

Пример ребуса:

$$\begin{array}{r}
 + \text{ОДИН} \\
 \text{ОДИН} \\
 \hline
 \text{МНОГО}
 \end{array}
 \qquad
 \begin{array}{r}
 + 6823 \\
 6823 \\
 \hline
 13646
 \end{array}$$

Лиза думает, что если ребус имеет решение, то слова, из которых он состоит, имеют скрытую связь. Она придумала уже целую кучу слов, которые она хочет проверить на наличие скрытой связи, и составила из них ребусы. Однако решение некоторых ребусов может занять очень много времени, поэтому Лиза просит вас помочь ей и написать программу, которая найдет решение для каждого ребуса или скажет, что его не существует.

Более формально:

Сопоставьте каждой из 3 строк S_1, S_2, S_3 , **положительные** числа N_1, N_2, N_3 , так что выполнено следующее:

- S_i имеет такую же длину, как N_i (в десятичной записи).
- В записи чисел N_1, N_2, N_3 нет ведущих нулей.
- Любым двум одинаковым буквам соответствуют одинаковые цифры, а любым двум разным буквам – разные цифры.

Формат входных данных

Даны 3 строки, состоящие из строчных латинских букв. Каждая строка имеет длину от 1 до 10.

Формат выходных данных

Если решения не существует – выведите **UNSOLVABLE**, иначе выведите три числа (каждое в отдельной строке) – ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
x y z	1 2 3
a b a	UNSOLVABLE
win lose game	530 6891 7421
four seven eight	UNSOLVABLE
love hate feel	4352 1872 6224

[C-B'] 1. Операции над точкой

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Вам даны строка T длины n , которая состоит только из символов **N**, **W**, **S**, **E**, и точка на плоскости. Сначала координаты точки равны $(0, 0)$. В течение следующих n дней вы совершаете операции над точкой. В день i вам нужно, в зависимости от символа T_i , сделать следующее:

- при $T_i = \mathbf{N}$, прибавить положительное вещественное число ко второй координате;
- при $T_i = \mathbf{W}$, отнять положительное вещественное число от первой координаты;
- при $T_i = \mathbf{S}$, отнять положительное вещественное число от второй координаты;
- при $T_i = \mathbf{E}$, прибавить положительное вещественное число к первой координате.

Для одного и того же направления в разные дни вы можете прибавлять (или отнимать) разные числа.

Сможете ли вы совершить операции таким образом, что после n дней координаты точки будут опять равны $(0, 0)$?

Формат входных данных

В единственной строке входных данных вам задана T ($1 \leq n \leq 1000$).

Формат выходных данных

Если возможно совершить описанные операции так, что в итоге координаты точки будут равны $(0, 0)$, то выведите «**Yes**» (без кавычек). Иначе выведите «**No**» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
WSEN	Yes
WEWWEW	Yes
WWSN	No
N	No

Замечание

В первом примере во все дни можно выбрать 1 в качестве положительного вещественного числа, что приведет к желаемому результату.

Во втором примере подходит прибавить к первой координате во второй день 4, в пятый день 12, а во все остальные отнять 4.

В последних двух примерах нельзя совершить операции так, чтобы в итоге координаты точки снова стали равны $(0, 0)$.

[C-B'] 2. Долгое обсуждение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно преподаватели одной широко известной в узких кругах смены решили обсудить заштрафные лекции. Как-то так вышло, что обсуждение лекции переросло в обсуждение всего, что происходит и не происходит в жизни. Так они просидели более двух часов, после чего обнаружили, что уже пора идти спать. Но случилось страшное — в учительской сломалась сигнализация, а потому нашим доблестным учителям надо взломать ее, чтобы иметь возможность спокойно покинуть учительскую.

На сигнализации были написаны целые числа n и k . Спустя час раздумий и минуту поиска в интернете, преподаватели узнали, что для взлома сигнализации необходимо посчитать количество троек целых чисел (a, b, c) таких, что выполняются следующие условия:

- $\min(a, b, c) \geq 1$
- $\max(a, b, c) \leq n$
- $(a + b) : k$ (то есть $(a + b)$ делится нацело на k)
- $(b + c) : k$
- $(a + c) : k$

Тройки чисел (a_1, b_1, c_1) и (a_2, b_2, c_2) считаются различными, если $a_1 \neq a_2$, $b_1 \neq b_2$ или $c_1 \neq c_2$. Например, тройки чисел $(1, 2, 3)$ и $(2, 1, 3)$ считаются различными.

Помогите бедным преподавателям покинуть учительскую, ведь иначе им придется спать там.

Формат входных данных

В первой строке даны целые числа n и k ($1 \leq n, k \leq 2 \cdot 10^5$).

Формат выходных данных

В единственной строке выведите количество троек целых чисел, удовлетворяющих условию выше.

Примеры

стандартный ввод	стандартный вывод
3 2	9
5 3	1
31455 777	64000
56788 922	465309

Замечание

В первом тесте подходят следующие тройки чисел:

- $(1, 1, 1)$
- $(2, 2, 2)$
- $(1, 1, 3)$
- $(1, 3, 1)$

- (3, 1, 1)
- (1, 3, 3)
- (3, 3, 1)
- (3, 1, 3)
- (3, 3, 3)

[C-B'] 3. Регулярка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дана строка, состоящая из строчных латинских букв, а так же символов `*` и `?`. Звездочка означает, что символ, стоящий перед ней, можно удалить, оставить или повторить любое количество раз. Знак вопроса – что символ, стоящий перед ним, можно либо же удалить, либо оставить.

Вам также дано число k . Можно ли из исходной строки получить некоторую строку длины k , которая не содержит звездочек и вопросительных знаков? Если да – приведите пример такой строки.

Формат входных данных

В первой строке записана строка, состоящая только из строчных латинских букв, а также символов `*` и `?`. Эти символы могут стоять **только сразу после буквы**. Длина строки не превосходит 200 символов.

Во второй строке находится целое число k ($1 \leq k \leq 200$) — необходимая длина сообщения.

Формат выходных данных

Выведите любую строку длины k , которую можно получить из данной строки, или `Impossible`, если такого сообщения не существует.

Примеры

стандартный ввод	стандартный вывод
g?o*dw?l?uckincontest 17	goodluckincontest
bd?c 2	bc
bd?c 3	bdc

[C-B'] 4. Трафареты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Михаил любит коллекционировать всякий хлам, особенно трафареты. Трафарет можно представить в виде набора из n строк, каждая из которых состоит из n символов «#» или «.». С трафаретом можно производить следующие операции:

1. Повернуть трафарет на 90 градусов по или против часовой стрелке относительно центра.
2. Сдвинуть трафарет вверх, если первая строка не содержит «#». Это значит, что вторая строка становится первой, третья — второй, ..., n -я — $(n - 1)$ -й, n -я строка будет состоять только из «.». Аналогично можно сдвинуть трафарет вниз, если n -я строка не содержит «#».
3. Сдвинуть трафарет влево, если первый столбец не содержит «#». Это значит, что второй столбец становится первым, третий — вторым, ..., n -й — $(n - 1)$ -м, n -й столбец будет состоять только из «.». Аналогично можно сдвинуть трафарет вправо, если n -й столбец не содержит «#».

Так как Михаил только начинающий хламосборщик, то у него есть дома только один трафарет A размера n . Недавно он нашел трафарет B такого же размера. Он попросил вас определить, являются ли эти два трафарета одинаковыми, если можно проделать любые операции выше несколько (возможно, ноль) раз.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 200$) — размер трафаретов.

В следующих n строках содержится описание трафарета A — строки длины n , состоящие из символов «#» или «.».

В следующих n строках содержится аналогичное описание трафарета B .

Формат выходных данных

Выведите «YES», если с помощью операций можно добиться совпадения трафаретов A и B , и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
<pre> 4 #... ..#.##. </pre>	YES
<pre> 3 ##. #.. .#. </pre>	NO

Замечание

В первом наборе входных данных можно повернуть трафарет A на 90 градусов по часовой стрелке, а потом сдвинуть трафарет B влево.

Можно показать, что во втором примере невозможно добиться совпадения трафаретов.

[C-B'] 5. Слишком много нот

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Петя составил мелодию из n нот. Для простоты будем считать, что в один момент времени может звучать только одна нота. Учитель по сольфеджио посмотрел на эту мелодию и сказал, что в ней «Слишком много нот». Петя расстроился, но все же послушался. Теперь ему надо уменьшить число различных используемых нот – для этого он хочет убрать все использования какой-то одной ноты, а все остальные оставить на своих местах относительно друг друга.

Но Петя хочет, чтобы полученная мелодия была гармоничной, поэтому просит вас удалить все вхождения одной ноты, чтобы получилась лексикографически наименьшая последовательность нот. Помогите Пете!

Ноты представлены в виде натуральных чисел в диапазоне от 1 до n .

Последовательность чисел A лексикографически меньше последовательности B , если A является началом B (например, $(1, 2) < (1, 2, 3)$) или если первый различающийся элемент в A меньше, чем в B (например, $(1, 2) < (1, 3)$).

Формат входных данных

В первой строке дано количество нот n ($1 \leq n \leq 200000$). В следующей строки записаны n нот – последовательность чисел, каждое из которых лежит в диапазоне от 1 до n .

Формат выходных данных

Выведите лексикографически наименьшую последовательность нот, которую можно получить удалением всех вхождений какой-то одной ноты.

Примеры

стандартный ввод	стандартный вывод
2 2 2	
5 1 2 1 2 1	1 1 1
5 4 5 5 3 4	4 3 4

Замечание

В первом примере можно удалить только двойку, тогда полученная последовательность будет пуста.

Во втором примере если удалить 1, то будет $(2, 2)$. А если удалить 2, то будет $(1, 1, 1)$. Наименьшая последовательность достигается при удалении двойки.

В третьем примере возможны варианты $(5, 5, 3)$ при удалении четверки, $(4, 3, 4)$ при удалении пятёрки и $(4, 5, 5, 4)$ при удалении тройки. Из трех вариантов наименьшим будет $(4, 3, 4)$.

[C-B'] 6. Хорошие были времена...

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дед Филипп уже не молод, и на своих лекциях вместо темы рассказывает, что в его годы грибы были краснее и проезд на автобусе дешевле. Дед жил в стародавние времена, когда в обиходе были только монеты номиналом 2, 3, 4, а проезд стоил всего 10 рублей.

И в очередной день, когда он забыл выпить таблетки, Филипп вспомнил, что ему на неделю давали a монет номинала 2, b монет номинала 3, c монет номинала 4.

Дед уже не молод, поэтому помогите ему посчитать сколько раз он мог проехать на автобусе на эту сумму денег, учитывая, что сдачу изобрели позже, а платить больше 10 рублей за проезд было нельзя.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

Единственная строка каждого набора входных данных содержит три целых числа a , b , c ($0 \leq a, b, c, \leq 10^{15}$) — количество монет номиналом 2, 3 и 4 соответственно.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — количество проездов на автобусе, которое можно совершить.

Пример

стандартный ввод	стандартный вывод
4	2
1 2 3	0
0 0 4	51
100 100 3	200
100 200 300	

[В'-В] 1. СОП

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В одном известном ПТУ практикуют студенческую оценку преподавания (СОП). По окончании курса каждый из n студентов ставит оценку преподавателю по шкале от 1 до m . Лектор Артём получил доступ к оценкам и решил удалить некоторые из них, чтобы порадовать себя. Максимальную радость Артём определяет как длину максимальной последовательности из одинаковых оценок (и ему всё равно, что они могут быть низкими!). Чтобы не вызывать подозрений, Артём может удалить не более d оценок. Помогите определить Артёму его максимальную радость. Обратите внимание, нельзя менять порядок оценок, можно удалять некоторые из них.

Формат входных данных

В первой строке входных данных содержится три целых числа n , m и d ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 10^5$, $0 \leq d < n$) — количество студентов, максимальную оценку на СОПе и количество удалений, которые не вызовут подозрений.

Во второй строке даны n целых чисел s_1, s_2, \dots, s_n ($1 \leq s_i \leq m$) — оценки, поставленные студентами.

Формат выходных данных

Выведите единственное число — максимально возможную радость Артёма.

Примеры

стандартный ввод	стандартный вывод
5 10 4 1 1 1 1 1	5
9 10 3 8 1 1 7 1 10 1 1 1	6
8 5 2 5 4 3 5 5 4 5 4	3

[В'-В] 2. Лабиринт в Майнкрафте

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Денис и Герман вспомнили детство, отрочество, юность и зачёт в параллели А. Поэтому им показалось хорошей идеей мысленно вернуться в 2013 год и пройти карту в игре Майнкрафт. Но когда они запустили локальный сервер на порту 40539, оказалось, что карта является лабиринтом...

Более конкретно, карта — это n комнат. Комнаты пронумерованы от 1 до n . В каждой комнате могут быть две двери: левая и правая. Проходя через дверь, ребята попадают в другую комнату. В этой другой комнате тоже могут быть двери, а ещё из нее можно вернуться обратно в предыдущую комнату по тому же ходу, через который в неё пришли. Ребята начинают в комнате с номером 1, из которой никуда нельзя вернуться. Более того, проходя через двери и обратные ходы, из любой комнаты можно дойти до любой другой, причем если не повторять комнаты на своём пути, то такой путь всегда единственен. Возможно, вы уже поняли, что лабиринт на самом деле представляется **деревом**.

Ребята выяснили, что выход из лабиринта появится в какой-то комнате, но в какой — заранее неизвестно. Более того, когда выход появится, в лабиринте выключится свет, а проходить через двери можно будет только 1 раз. Поэтому они решили заранее узнать структуру лабиринта. Для этого они воспользовались простой стратегией:

1. Если в комнате есть левая дверь, через которую они ещё не ходили, то шли через неё;
2. Иначе если в комнате есть правая дверь, через которую они ещё не ходили, то шли через неё;
3. Иначе они возвращались через обратный ход в предыдущую комнату, если он есть.

В конце ребята вернулись в комнату с номером 1 и закончили исследовать лабиринт. Конечно, просто погулять по лабиринту интересно, но нужно было куда-то записывать его структуру. Для этого у ребят были книги. К сожалению, поля книг были слишком малы, чтобы нарисовать в них лабиринт явно. Поэтому игроки выписывали номера комнат в некотором порядке. Денис выписывал номер комнаты каждый раз, когда они попадали в неё впервые. Независимо от него Герман выписывал номер, если

- они находятся в соответствующей комнате;
- он ещё не выписывал соответствующий номер;
- если в комнате есть левая дверь, то через левую дверь в этой комнате они уже должны пройти (если в комнате не было левой двери, то Герман это условие игнорировал).

Когда ребята вернулись, у них получилось две перестановки номеров комнат. Игровая карта была такова, что «выходом» было исчезновение пола и попадание в лаву, после которого игроки респавнились в «победной комнате». Конечно, ребята догадались, что они могут не проходить лабиринт и прокопали блоки под собой. После этого случая автор карты включил всем новым игрокам *adventure mode* и не дает исследовать карту. Теперь вам, чтобы пройти лабиринт, нужно восстановить его структуру. Вам очень повезло, ведь вам в руки попались перестановки, которые выписали Денис и Герман. Восстановите структуру лабиринта или скажите, что ребята ошиблись.

Формат входных данных

На первой строке находится единственное число n ($2 \leq n \leq 2 \cdot 10^5$) — количество комнат в лабиринте. На второй строке находится перестановка из чисел от 1 до n — номера комнат в порядке, в котором их выписал Денис. На третьей строке находится перестановка из чисел от 1 до n — номера комнат в порядке, в котором их выписал Герман.

Формат выходных данных

Если не существует **ровно одного** лабиринта, в котором ребята могли получить данные вам перестановки, начав обход лабиринта в комнате с номером 1, то выведите на единственной строке число -1 .

Если же такой лабиринт существует, то выведите n строк. На i -й строке должно находиться два числа x_i и y_i , означающие, что левая дверь из комнаты i ведёт в комнату x_i , а правая дверь ведёт в комнату y_i . Если какая-то дверь в комнате отсутствует, то вместо соответствующего номера (x_i или y_i) выведите число 0.

Примеры

стандартный ввод	стандартный вывод
6 1 3 5 6 4 2 3 5 1 4 6 2	3 6 0 0 0 5 0 0 0 0 4 2
2 2 1 1 2	-1

Замечание

Лабиринт, описанный в первом тесте, выглядит следующим образом:



Направления «лево» и «право» соответствуют рисунку

[В'-В] 3. Никита собирает апельсины

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как известно, Никита очень любит апельсины. Мир, в котором живет Никита, представляет собой числовую прямую, на которой расположены n деревьев с апельсинами.

Дерево с номером i имеет координату x_i , и апельсины на нем созревают в момент времени t_i . Известно, что все координаты деревьев попарно различны.

Изначальная координата Никиты равна 0. Двигаясь на одну единицу по числовой прямой, у Никиты копится одна единица гнева.

Никита очень хочет кушать, поэтому в каждый момент времени, когда созревают новые апельсины, Никита **должен** их всех съесть. Никита настолько голоден, что передвижение по числовой прямой не занимает у него времени, а только копит гнев.

Найдите минимальный итоговый гнев Никиты, чтобы съесть все апельсины и **вернуться** в координату 0.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 200\,000$) — количество деревьев с апельсинами.

Следующие n строк содержат два целых числа x_i и t_i ($-10^9 \leq x_i \leq 10^9$, $1 \leq t_i \leq n$, $x_i \neq 0$) — координата i -го дерева и момент созревания на нем апельсинов.

Гарантируется, что в координате 0 нет дерева и что все координаты деревьев попарно различны.

Формат выходных данных

Выведите минимальный итоговый гнев Никиты, когда он съест все апельсины и вернется в изначальную координату.

Пример

стандартный ввод	стандартный вывод
5 2 1 1 4 5 1 4 4 3 2	12

Замечание

Рассмотрим пример из условия. Один из оптимальных маршрутов Никиты выглядит так:

- В момент времени 0 Никита находится в координате 0, и накопил 0 гнева.
- В момент времени 1 появляются апельсины в координатах 2 и 5. Никита съедает сначала апельсин в координате 2, а затем в координате 5. В итоге Никита оказывается в координате 5, и он накопил 5 гнева.
- В момент времени 2 появляется апельсин в координате 3. Никита идет за ним и съедает, теперь он в координате 3, и у него 7 единиц гнева.
- В момент времени 4 появляются апельсины в координатах 1 и 4. Никита сначала идет за апельсином в координате 4, затем за апельсином в координате 1, и затем возвращается в координату 0. Итого, Никита накопил $7 + (4 - 3) + (4 - 1) + (1 - 0) = 7 + 1 + 3 + 1 = 12$ единиц гнева.

[В'-В] 4. Это конец

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Михаил общается со своей очень хорошей подругой. Их общение длится n дней, каждый из которых можно представить в виде последовательности из открывающих и закрывающих круглых скобок. Их общение находится на грани разрыва, а поэтому вы (так как вы с ним похожи на братьев) должны спасти его.

Общение Михаила будет спасено в том и только в том случае, если возможно сконкатенировать эти n последовательностей в произвольном порядке так, чтобы в итоге получилась правильная скобочная последовательность. Это связано с тем, что тогда Михаил сможет доказать своей подруге, что он всегда пишет первым, да и к тому же столько же, сколько и она. Если не существует такого порядка конкатенации, то вы должны сообщить эту грустную новость Михалу, чтобы он подготовился к длительным грустным дням.

Напомним, что правильными скобочными последовательностями называются следующие последовательности скобок:

- $()$,
- (A) , если A является правильной скобочной последовательностью,
- AB , если A и B являются правильными скобочными последовательностями.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^6$) — количество последовательностей.

Каждая из следующих n строк содержит непустую строку, состоящую из открывающих и закрывающих круглых скобок. Строки пронумерованы целыми числами от 1 до n в порядке следования. Заметьте, что эти строки не обязаны являться правильными скобочными последовательностями.

Гарантируется, что суммарная длина всех строк не превосходит 10^7 символов.

Формат выходных данных

Если спасти общение Михаила возможно, то выведите перестановку чисел от 1 до n — в каком порядке надо конкатенировать скобочные последовательности, чтобы в итоге получить правильную скобочную последовательность.

Если же Михаил обречен (то есть не существует требуемого порядка), то выведите число «-1» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
2 ((())())()	1 2
5)) (((())	2 3 4 5 1
3 ((((((-1

[В'-В] 5. Операция над строкой

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Дана строка $S = S_1S_2 \dots S_n$ из строчных символов английского алфавита. Вы можете сделать с ней следующее преобразование: выбрать индексы i и j ($1 \leq i \leq j \leq n$) и развернуть подстроку $S_iS_{i+1} \dots S_j$ (поменять в исходной строке ее на $S_jS_{j-1} \dots S_i$). Сколько различных строк можно получить, применив такую операцию не больше одного раза?

Формат входных данных

Во входных данных дана строка S ($1 \leq |S| \leq 200\,000$). Она состоит только из строчных букв английского алфавита.

Формат выходных данных

В качестве ответа выведите единственное целое число – количество возможных различных строк.

Примеры

стандартный ввод	стандартный вывод
ххуу	5
aaaaaaaaa	1
bcbabcb	15

Замечание

В первом примере достижима строки «ххуу» (не надо делать преобразований), «хуху» (надо развернуть $S[2..3]$), «хуух» (надо развернуть $S[2..4]$), «уухх» (надо развернуть $S[1..4]$) и «ухху» (надо развернуть $S[1..3]$).

Во втором примере всегда получается «aaaaaaaaa».

[В'-В] 6. Работа дворником

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

К черту это спортивное
программирование! Пойду работать
дворником!

Михаил

Однажды, после неудачно написанного контеста на Codeforces, Михаил окончательно разочаровался в своей карьере спортивного программиста и решил устроиться на работу дворником. Однако, даже на такую должность его ждало собеседование.

В качестве собеседования Михаилу предложили очистить от мусора довольно большой двор. Для удобства представим двор как прямоугольную таблицу, состоящую из n строк и m столбцов. Строки пронумерованы сверху вниз числами от 1 до n , а столбцы — слева направо числами от 1 до m . На пересечении i -й строки и j -го столбца находится клетка с координатами (i, j) .

Изначально в каждой клетке двора лежит огромная куча мусора. Михаил в рамках своего собеседования произвел k операций, каждая из которых была одного из двух типов:

- Убрать весь мусор во всех клетках (i, j) , таких что $i + j = s$;
- Убрать весь мусор во всех клетках (i, j) , таких что $i - j = d$.

Ваша задача — определить количество клеток, в которых Михаил убрал весь мусор.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^9$) — размеры двора.

Вторая строка содержит одно целое число k ($0 \leq k \leq 300\,000$) — количество операций, совершенных Михаилом.

Каждая из следующих k строк содержит описание очередной операции, совершенной Михаилом в следующем формате:

- $+ \ s$ ($2 \leq s \leq n + m$) — данная операция означает, что Михаил убрал весь мусор во всех клетках (i, j) , таких что $i + j = s$;
- $- \ d$ ($1 - m \leq d \leq n - 1$) — данная операция означает, что Михаил убрал весь мусор во всех клетках (i, j) , таких что $i - j = d$.

Формат выходных данных

Выведите одно целое число — количество клеток, в которых Михаил убрал весь мусор.

Пример

стандартный ввод	стандартный вывод
4 4 4 + 4 + 2 - 0 - -3	7

Замечание

Рассмотрим действия Михаила в примере из условия:

1. Михаил убирает мусор из клеток (i, j) , для которых $i + j = 4$. Это клетки с координатами $(1, 3)$, $(2, 2)$ и $(3, 1)$.
2. Михаил убирает мусор из клеток (i, j) , для которых $i + j = 2$. Это клетка с координатами $(1, 1)$.
3. Михаил убирает мусор из клеток (i, j) , для которых $i - j = 0$. Это клетки с координатами $(1, 1)$, $(2, 2)$, $(3, 3)$ и $(4, 4)$.
4. Михаил убирает мусор из клеток (i, j) , для которых $i - j = -3$. Это клетка с координатами $(1, 4)$.

Таким образом, мусор был убран из клеток $(1, 1)$, $(1, 3)$, $(1, 4)$, $(2, 2)$, $(3, 1)$, $(3, 3)$ и $(4, 4)$.

[В-А'] 1. Загадка Жака Фреско

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Однажды, сидя на берегу океана, размышляя о вечном, вы внезапно встретили своего давнего друга, который очень любит теорию чисел. Он сел рядом с вами и произнес следующее: «было два числа A и B , сколько?» – после чего он ушел. Вы его очень давно знаете, поэтому сразу поняли, что он имел в виду.

A именно, есть два целых положительных числа A и B . Рассмотрим произведение всех делителей числа A^B , пусть это значение равно C . Тогда сколько раз число C можно поделить на A нацело?

Поскольку ответ может быть очень большим, пожалуйста, выведите его по модулю 998244353.

Формат входных данных

Единственная строка входных данных содержит два целых числа A , B ($2 \leq A \leq 10^{12}$, $0 \leq B \leq 10^{18}$).

Формат выходных данных

Сколько?

Примеры

стандартный ввод	стандартный вывод
2 0	0
5 2	3
5252 5252	688115020
4 4	18

[В-А'] 2. Змейка 2007

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Никто никогда не вернется в 2007. Но у вас получилось это сделать на небольшое время.

Конечно же, первым делом вы взяли телефон Nokia 3310, чтобы поиграть в змейку. К сожалению, прямо в процессе игры он сломался (как известно, этот телефон сломать невозможно).

Экран больше ничего не показывает, змейка остановилась. Однако вы заметили, что если надавливать на прямоугольные области экрана, можно узнавать информацию о змейке.

Формально экран это клетчатая таблица размера $n \times n$. В этой таблице располагается ровно одна змейка: некоторый путь от головы змейки до хвоста, соседние клетки пути являются соседями по стороне. Гарантируется, что хвост и голова располагаются в разных клетках. Все клетки пути различные.

Если надавить на прямоугольную область экрана, то вы увидите, сколько раз путь пересекал границу области (вы уверены в своих силах верно вычислять это число каждый раз). Формально количество пересечений это количество пар соседних клеток пути, таких что одна из них лежит внутри области, а другая снаружи.

Используя не более 2007 надавливаний, узнайте, в каких клетках находятся хвост и голова змейки (в любом порядке).

Протокол взаимодействия

В первой строке находится единственное целое число n ($2 \leq n \leq 1000$) — размер экрана.

Далее не более 2007 раз вы можете делать запрос надавливания на экран.

Чтобы сделать запрос, выведите ? r_1 c_1 r_2 c_2 ($1 \leq r_1 \leq r_2 \leq n$, $1 \leq c_1 \leq c_2 \leq n$). Внутри этого прямоугольника лежат все клетки (r, c) , такие что $r_1 \leq r \leq r_2$, $c_1 \leq c \leq c_2$.

После каждого запроса не забудьте вывести перевод строки и сбросить буфер вывода.

В качестве ответа на запрос вам будет дано единственное целое число — количество пересечений змейки с границей области. Если вы сделали некорректный запрос или превысили лимит на количество запросов, вам будет выведено число -1 . Ваша программа должна корректно завершиться в этом случае, иначе вердикт решения может быть любым.

Когда вы готовы вывести ответ, выведите ! r_1 c_1 r_2 c_2 , где (r_1, c_1) и (r_2, c_2) это две клетки, в которых находятся голова и хвост (можно вывести в любом порядке).

Гарантируется, что интерактор не является адаптивным.

Пример

стандартный ввод	стандартный вывод
4	? 3 3 4 4
1	? 1 4 1 4
0	? 2 1 3 4
3	? 1 1 4 3
2	! 3 3 1 1

Замечание

В примере змейка состоит из 9 клеток:

$$(1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow (3, 2) \rightarrow (4, 2) \rightarrow (4, 3) \rightarrow (4, 4) \rightarrow (3, 4) \rightarrow (3, 3)$$

[В-А'] 3. Рундук

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Гуляя по необычному лесу Петя нашел необычную речку, на берегу которой лежал рундук — сундук для рун, а рядом стоял гномик. Рундук оказался закрытым на необычный замок. У гномика Петя получил два массива a и b длины n . Замок представляет из себя таблицу из n строк и n столбцов. Оказалось, что на пересечении i строки и j столбца находится число $a_i + b_j$.

Петя очень хочет открыть этот рундук, для этого нужно ответить на q запросов от гномика. Запрос описывается четырьмя целыми числами x_1, x_2, y_1, y_2 — верхний левый и правый нижний угол подтаблицы замка, ответом на запрос служит \gcd всех чисел в данной подтаблице. Петя хочет как можно быстрее открыть рундук, но не имеет вайба думать над загадками.

Помогите ему и ответьте на все запросы гномика!

Формат входных данных

В первой строке содержится два целых числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) — длины массивов и количество запросов.

Во второй строке содержится n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — описание массива a .

В третьей строке содержится n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^9$) — описание массива b .

В следующих q строках содержится 4 целых числа x_1, x_2, y_1, y_2 ($1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq n$) — описание запроса.

Формат выходных данных

Выведите q строк, в каждой единственное целое число — ответ на запрос.

Примеры

стандартный ввод	стандартный вывод
3 5	2
3 5 6	2
1 7 3	4
1 2 1 2	1
1 2 2 3	2
2 2 2 3	
1 3 1 3	
1 1 1 2	
2 3	1337
442 26	52
895 26	52
1 1 1 1	
2 2 2 2	
1 2 2 2	

[В-А'] 4. Карта торгового центра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

После нового видео блогера, который запутался в торговом центре (даже имея карту), администрация решила видоизменить структуру здания. Торговый центр представляет из себя n галерей с двусторонними переходами между ними. Чтобы изменить структуру администрация решила выбрать k пронумерованных секторов, состоящих из каких-то галерей. Каждый сектор должен состоять из хотя бы одной галереи, и каждая галерея должна состоять в каком-то секторе. Обозначим номер сектора в котором состоит i -я галерея как p_i .

Чтобы блогеры больше не путались было принято решения выбирать сектора таким образом, чтобы все переходы соединяли галереи только из соседних секторов. То есть для каждого перехода который соединяет галереи (i, j) должно выполняться условие $|p_i - p_j| = 1$. Обратите внимание, что переходы не могут соединять галереи, которые состоят в одном и том же секторе.

Чтобы в торговом центре было больше контента принято решение найти максимально возможное количество секторов k . Так как торговый центр оказался и правду сложным, администрация не смогла решить эту задачу самостоятельно, помогите им!

Формат входных данных

В первой строке содержится единственное число n ($2 \leq n \leq 200$) — количество галерей в изначальной карте торгового центра.

В следующих n строках содержится описание переходов между галереями, описание представляет собой таблицу S , которая состоит из символов 0 и 1. Где $S_{i,j}$ обозначает наличие перехода между галереями i и j . Гарантируется, что от каждой галереи можно добраться до каждой другой галереи через один или несколько переходов. Гарантируется, что не существует переходов внутри галереи ($S_{i,i} = 0$ для каждого $1 \leq i \leq n$).

Формат выходных данных

В единственной строке выведите единственное целое число — ответ на задачу. Если ни один сектор выделить не получается выведите -1.

Примеры

стандартный ввод	стандартный вывод
3 010 101 010	3
4 0111 1011 1101 1110	-1
6 010011 100001 000110 001001 101000 110100	-1

[В-А'] 5. Опустошение канистры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно, наводя порядок у себя в подвале, вы наткнулись на канистру, целиком заполненную водой. Так как канистра была сделана в СССР, ее внешний вид оставляет желать лучшего. Для удобства представим канистру как многоугольник на плоскости, внутренность которого заполнена водой.

Вы хотите вылить всю воду из канистры. Для этого вы можете просверлить некоторое количество отверстий в канистре. Иными словами, вы можете просверлить отверстие в произвольной точке на границе многоугольника.

Ваша задача — просверлить как можно меньше отверстий таким образом, чтобы вся вода вылилась из канистры.

Будем считать, что вода является идеальной жидкостью, которая по возможности стремится течь вниз (иными словами, она всегда течет в сторону уменьшения y -координаты). В частности, если вода находится над горизонтальной поверхностью, в которой просверлено отверстие, вся жидкость вытечет через него. Для лучшего понимания физики воды обратите внимание на примеры.

Формат входных данных

Первая строка содержит одно целое число n ($3 \leq n \leq 2\,000$) — количество вершин многоугольника.

Каждая из следующих n строк содержит два целых числа x_i и y_i ($|x_i|, |y_i| \leq 10\,000$) — координаты i -й вершины многоугольника.

Вершины многоугольника заданы в порядке обхода против часовой стрелки. Многоугольник является простым. Это означает, что вершины многоугольника попарно различны, а также никакие две стороны не пересекаются (кроме сторон, являющихся соседними в порядке обхода). **Не гарантируется**, что последовательные стороны многоугольника не лежат на одной прямой.

Формат выходных данных

Выведите одно целое число — минимальное количество отверстий, которые нужно просверлить.

Примеры

стандартный ввод	стандартный вывод
3 0 0 2 0 1 1	1
4 0 0 1 -1 2 0 1 1	1
6 0 0 1 -1 2 0 3 -1 4 0 2 10	2

Замечание

В первом примере канистра имеет форму треугольника. Достаточно просверлить одно отверстие в любой точке на стороне между вершинами $(0, 0)$ и $(2, 0)$.

Во втором примере достаточно просверлить одно отверстие в точке $(1, -1)$.

В третьем примере достаточно просверлить два отверстия в точках $(1, -1)$ и $(3, -1)$.

[В-А'] 6. Смена жилья

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Берляндии наступил новый учебный год! Разумеется, первокурсникам пора переезжать и начинать самостоятельную жизнь.

Всего в Берляндии присутствует n домов. Так совпало, что в этом году первокурсников тоже ровно n , причем i -й первокурсник живет в доме под номером i .

В этом году честь распределять новых студентов по квартирам выпала Вам. Вы можете для каждого первокурсника определить дом, в котором будет располагаться квартира, в которой он будет жить. При этом правила распределения запрещают Вам селить двух студентов в одном доме.

Каждый человек обладает некоторой привязанностью к семье. i -й первокурсник имеет a_i единиц привязанности к семье. Если i -го студента переселить в дом под номером j , то его тоска по дому будет иметь значение $a_i \cdot |i - j|$.

Вы решили немножечко пошутить. А именно Вы решили расселить первокурсников таким образом, чтобы **максимизировать** их суммарную тоску. Определите, какого максимального значения суммарной тоски Вы можете достичь.

Формат входных данных

Первая строка содержит единственное число n ($2 \leq n \leq 2000$) — количество первокурсников в новом учебном году.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n , где a_i ($1 \leq a_i \leq 10^9$) — уровень привязанности i -го первокурсника к семье.

Формат выходных данных

Выведите одно число — максимальное суммарное значение тоски по дому, которого можно достичь, перераспределив первокурсников по домам.

Примеры

стандартный ввод	стандартный вывод
4 1 5 2 3	23
6 1 2 3 4 5 6	67

Замечание

В первом примере можно поступить следующим образом:

- первого студента поселить в дом 3. Его тоска составит $1 \cdot |1 - 3| = 2$.
- второго студента поселить в дом 4. Его тоска составит $5 \cdot |2 - 4| = 10$.
- третьего студента поселить в дом 2. Его тоска составит $2 \cdot |3 - 2| = 2$.
- четвертого студента поселить в дом 1. Его тоска составит $3 \cdot |4 - 1| = 9$.

Общая тоска составит $2 + 10 + 2 + 9 = 23$.

[А'-А] 1. Оля и НВП

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно Оля нашла нового друга. На радостях она почти опоздала на поезд, но, к счастью, есть ещё несколько минут... Контролёр выдал Оле массив a длины n . Чтобы сесть на поезд, ей нужно поменять не более одного элемента массива на какое-то произвольное целое число (например, на -57^{179}) так, чтобы максимизировать длину НВП (наибольшей возрастающей подпоследовательности).

Оля с лёгкостью справилась с этой задачкой. А получится ли у вас?

Формат входных данных

В первой строке ввода находится единственное целое число n — длина массива a ($1 \leq n \leq 2 \cdot 10^5$).
Во второй строке находятся n целых чисел — элементы массива a ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите единственное число — максимальную возможную длину НВП после изменения.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 3 5	5
5 1 2 3 3 4	4

Замечание

В первом примере можно заменить a_4 на 4, чтобы получить массив с НВП длины 5.
Во втором примере улучшить исходный ответ невозможно.

[A'-A] 2. Ctrl+F игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вы придумали новую игру под названием «Ctrl+F игра»

В начале игры вы открываете в браузере некоторую страницу с текстом, после чего открываете поиск вхождений на странице (с помощью комбинации Ctrl+F). Текст страницы это последовательность символов t_1, t_2, \dots, t_n .

В начале игры в поле вводится некоторый текст, являющийся подстрокой текста таблицы. Вы можете совершать ходы, за каждый ход можно либо удалить либо дописать произвольный символ в начало или конец текста, который сейчас находится в поле поиска. При этом текст в поле должен оставаться непустым и после хода текущий текст в поле должен оставаться подстрокой текста страницы t .

Вам даны два текста a_1, a_2, \dots, a_m и b_1, b_2, \dots, b_k , которые являются подстроками текста страницы t . За какое минимальное количество ходов можно получить из первого текста второй, совершая корректные ходы? Несложно убедиться, что получить второй текст из первого всегда возможно.

Формат входных данных

В первой строке находится единственное целое число n ($1 \leq n \leq 200\,000$) — длина текста страницы.

Во второй строке находится n целых чисел t_1, t_2, \dots, t_n ($1 \leq t_i \leq n$) — текст страницы (числа обозначают символы).

Во третьей строке находится единственное целое число m ($1 \leq m \leq n$) — длина первого текста.

В четвертой строке находится m целых чисел a_1, a_2, \dots, a_m . Гарантируется, что a является подстрокой t .

В пятой строке находится единственное целое число k ($1 \leq k \leq n$) — длина второго текста.

В шестой строке находится k целых чисел b_1, b_2, \dots, b_k . Гарантируется, что b является подстрокой t .

Формат выходных данных

Выведите единственное целое число — минимальное количество ходов, которые нужно сделать, чтобы получить из текста a текст b , совершая корректные ходы по правилам игры.

Примеры

стандартный ввод	стандартный вывод
5 1 1 2 3 1 4 1 2 3 1 2 1 1	4
9 1 2 2 5 6 7 3 4 3 3 3 4 3 3 1 2 2	12

[А'-А] 3. Минимальный диск

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Маша нарисовала на плоскости n отрезков (в рамках этой задачи концы отрезков являются их частью). Проанализировав рисунок Маши, Лера поняла, что никакие два отрезка не пересекаются. Женья решил нарисовать на плоскости круг (т.е. окружность со внутренностью), который пересекал бы все n отрезков. Чтобы всё не было так просто, Витя предложил искать не произвольный круг, а круг наименьшего возможного радиуса.

Ребята очень устали, пока обсуждали всю эту геометрию, поэтому разбираться с тем, как всё-таки решать задачу, придётся вам.

Формат входных данных

В первой строке ввода находится единственное целое число n — количество отрезков ($2 \leq n \leq 100$).

Каждая из следующих n строк содержит четыре целых числа a_i, b_i, c_i, d_i — координаты концов i -го отрезка ($0 \leq a_i, b_i, c_i, d_i \leq 1000$).

Гарантируется, что все отрезки имеют ненулевую длину и что никакая пара отрезков не пересекается.

Формат выходных данных

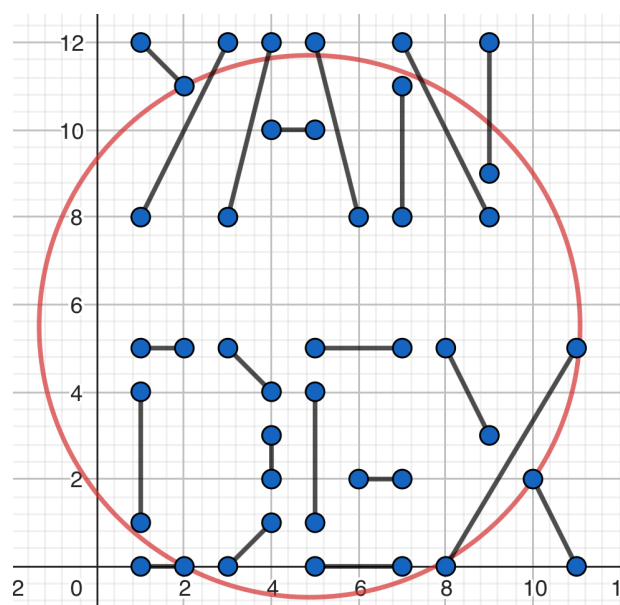
Выведите единственное вещественное число — наименьший возможный радиус круга. Ответ будет считаться правильным, если абсолютная или относительная ошибка не превосходит 10^{-5} .

Пример

стандартный ввод	стандартный вывод
21 1 8 3 12 1 12 2 11 3 8 4 12 4 10 5 10 5 12 6 8 7 8 7 11 7 12 9 8 9 9 9 12 3 5 4 4 4 2 4 3 4 1 3 0 2 0 1 0 2 5 1 5 1 1 1 4 5 0 7 0 5 1 5 4 5 5 7 5 6 2 7 2 8 0 11 5 8 5 9 3 10 2 11 0	6.2061

Замечание

Ниже приведено расположение отрезков в примере и оптимальный круг, центр которого расположен в точке $(4.875, 5.5)$



[А'-А] 4. Левин, Правин и два массива

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Два друга Левин и Правин получили на Новый год массивы — оба длины n . Левин массив a , Правин массив b . Левин с детства любит минимумы, Правин уверенно любит сумму.

Вместе они придумали необычное число x и стали искать разные статистики на своих массивах. Левин искал количество подотрезков минимум на которых не больше x , Правин искал количество подотрезков с суммой не больше x . Каждый нашел количество подотрезков довольно быстро, поэтому они решили объединить две задачи в одну (длины массивов же одинаковые). То есть найти количество подотрезков (l, r) , что $\min(a_l, a_{l+1}, \dots, a_r) + b_l + b_{l+1} + \dots + b_r \leq x$. Каждый посчитал ответ, но значения не сошлись, поэтому снова начались споры.

Разрешите проблему Левина и Правина — найдите количество таких подотрезков.

Формат входных данных

В первой строке содержатся два целых числа n, x ($1 \leq n \leq 2 \cdot 10^5, 0 \leq x \leq 3 \cdot 10^{14}$) — длины массивов и необычное число.

Во второй строке содержится n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^{14}$) — описание массива a .

В третьей строке содержится n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$) — описание массива b .

Формат выходных данных

В единственной строке выведите целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
4 12 4 5 1 3 2 7 3 4	6
12 28 3 4 2 2 8 1 4 1 1 5 10 8 3 7 9 2 4 2 8 1 8 4 9 6	52
12 30 10 5 1 10 5 9 1 3 4 8 8 8 4 2 8 2 5 3 5 2 6 9 3 5	57

[А'-А] 5. Клонирование графов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как ни странно, у клонов день рождения в один день, поэтому всем k клонам на день рождения подарили графы. Изначально у i клона был свой неориентированный граф G_i с n_i вершинами и m_i ребрами. Но клоны не были бы клонами, если бы они не клонировали все свои графы.

Поскольку клоны устали от того, что при клонировании получается много объектов, они решили клонировать все свои графы в один большой неориентированный граф G . А именно, они создали новый неориентированный граф с $n_1 \cdot n_2 \cdot \dots \cdot n_k$ вершинами, причем каждая вершина пронумерована каким-то массивом длины k (v_1, v_2, \dots, v_k) , где $1 \leq v_i \leq n_i$. Между двумя вершинами (v_1, v_2, \dots, v_k) и (u_1, u_2, \dots, u_k) есть ребро только в том случае, если для каждого i между v_i и u_i есть ребро в графе G_i .

Клоны только получили граф G , поэтому они хотят посчитать на нем какие-то базовые значения. А именно, им интересно суммарное расстояние от вершины $(1, 1, \dots, 1)$ до всех вершин, достижимых из $(1, 1, \dots, 1)$. Расстоянием между двумя вершинами называется минимальное количество ребер в пути от одной вершины до другой.

Поскольку ответ может быть очень большим, выведите ответ по модулю $10^9 + 7$.

Формат входных данных

В первой строке входных данных содержится целое число k ($2 \leq k \leq 5 \cdot 10^4$) – количество клонов. Далее следует описание k графов.

В первой строчке описания i графа содержится два целых числа n_i и m_i ($2 \leq n_i \leq 10^5, n_i - 1 \leq m_i \leq 2 \cdot 10^5$) – количество вершин и ребер в i графе соответственно.

В следующих m_i строчках содержатся два целых числа u, v ($1 \leq u, v \leq n_i$) – концы очередного ребра в i графе.

Гарантируется, что каждый граф не содержит кратных ребер, но могут быть петли.

Гарантируется, что сумма n_i не превосходит 10^5 , а также сумма m_i не превосходит $2 \cdot 10^5$.

Формат выходных данных

В единственной строке выведите ответ на задачу по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
<p>2</p> <p>4 4</p> <p>1 2</p> <p>2 3</p> <p>3 4</p> <p>4 1</p> <p>3 2</p> <p>3 1</p> <p>2 3</p>	<p>8</p>
<p>3</p> <p>4 6</p> <p>1 2</p> <p>1 3</p> <p>1 4</p> <p>2 3</p> <p>2 4</p> <p>3 4</p> <p>5 7</p> <p>1 3</p> <p>1 4</p> <p>1 5</p> <p>2 3</p> <p>2 4</p> <p>3 4</p> <p>5 2</p> <p>2 2</p> <p>1 2</p> <p>2 2</p>	<p>74</p>

[А'-А] 6. Горы красной поляны

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Вы планируете поехать на красную поляну, чтобы полюбоваться красотами и покататься на фуникулерах. Предвкушая поездку, вы решили порешать задачи.

В горном хребте находится n гор, расположенных в ряд слева направо. Вы знаете, что высота i -й горы является целым числом от 1 до u_i (включительно), однако не знаете точную высоту h_i .

Во время поездки вы будете забираться на все горы от 1 до n и на вершине каждой горы смотреть на пейзаж слева. Обозначим за a_i номер горы, на которую будет направлен ваш взгляд, то есть максимальный индекс $1 \leq j < i$, такой что $h_j > h_i$. Если таких индексов не существует, то $a_i = -1$. Назовем массив a профилем горного хребта.

Сколько существует различных профилей для всех возможных горных хребтов, подходящих под ограничения высот гор? Поскольку это число может быть слишком большим, найдите его по модулю 1 000 000 007.

Формат входных данных

В первой строке находится единственное целое число n ($1 \leq n \leq 100$) — количество гор.

Во второй строке находится n целых чисел u_1, u_2, \dots, u_n ($1 \leq u_i \leq 100\,000$) — ограничения сверху на высоты гор.

Формат выходных данных

Выведите единственное целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 2 1 3	3
5 4 3 1 5 3	21
6 100000 10000 1000 100 10 1	65

[А] 1. Летний снег

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Не очень тёплым летним днём Андрей вышел на бесконечную в обе стороны улицу и увидел снег... Не растерявшись, он разбил улицу на отрезки длиной один метр и сделал замеры глубины сугробов. Оказалось, что снег выпал равномерно по всей улице, и глубина сугроба везде составляет 10^9 нанометров.

Андрей решил закончить этот беспредел и растопить весь снег! Повезло, что он не выходит из дома без обогревателя в левом кармане. Андрей обратил внимание, что если включить обогреватель на одну секунду, то глубина сугроба на отрезке улицы, на котором он стоит, уменьшится на один нанометр. Обогреватель очень мощный, поэтому в правом кармане Андрей всегда носит вентилятор, чтобы использовать его в случае перегрева. Он выяснил, что если включить вентилятор, то глубина сугроба на отрезке улицы, на котором он стоит, увеличится на один нанометр.

Андрей начал блуждать по улице, периодически используя обогреватель и вентилятор. Его действия можно описать строкой $s_1 \dots s_n$ длины n , где s_i — это действие, произошедшее в i -ю секунду. Действия бывают четырёх типов:

- «<» означает, что Андрей переместился на отрезок улицы слева от того, на котором он стоял.
- «>» означает, что Андрей переместился на отрезок улицы справа от того, на котором он стоял.
- «-» означает, что Андрей включил обогреватель на одну секунду.
- «+» означает, что Андрей включил вентилятор на одну секунду.

Вы заметили, что Андрей мог совершить много лишних действий, поэтому вы захотели найти количество отрезков $1 \leq l \leq r \leq n$, что если бы Андрей выполнил только действия $s_l \dots s_r$ в таком порядке, то глубина сугробов на каждом отрезке улицы стала бы такой же, как после блуждания Андрея. Обратите внимание, что Андрей не может выбирать, с какого отрезка улицы начинать блуждание. Таким образом, начать блуждание он может только с фиксированного заранее отрезка улицы.

Формат входных данных

Первая строка содержит единственное число n ($1 \leq n \leq 250\,000$) — количество действий Андрея.

Вторая строка содержит строку $s_1 \dots s_n$ ($s_i \in \{<, >, -, +\}$) — действия Андрея.

Формат выходных данных

Выведите одно число — количество отрезков $1 \leq l \leq r \leq n$, удовлетворяющих критерию из условия.

Примеры

стандартный ввод	стандартный вывод
4 -+<>	6
8 <<<<>>>>	36
13 ->+<+>-<-<+>-	2

Замечание

В третьем примере после блуждания Андрея глубина сугроба, с которого он начинал, уменьшилась на 2. Глубина соседнего слева сугроба увеличилась на 1. Если бы Андрей выполнил операции «-<+>-», то изменение глубин было бы таким же. Таким образом, отрезок $l = 9$, $r = 13$ подходит под критерий. Вторым подходящим под критерий отрезком является $l = 1$, $r = 13$.

[A] 2. Dota stars

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Кирилл решил бросить спортивное программирование и заняться киберспортом. Его любимой игрой является «dota stars», в которой есть n героев, пронумерованных от 1 до n . Кирилл составил себе план тренировок для улучшения навыков игры за каждого из героев. Для этого ему надо составить некоторое количество команд так, чтобы i -й герой входил хотя бы в a_i команд.

Команда — это последовательное множество героев. Иными словами, команда однозначно задаётся двумя числами $1 \leq l \leq r \leq n$ и содержит героев с номерами от l до r включительно ровно по одному разу. Хотя Кирилл и любит сильные команды, в этот раз он решил улучшить свои навыки, играя за слабые команды. Силу команды из героев с номерами от l до r он считает равной $\max(p_l, \dots, p_r)$, где p_i — сила i -го героя.

Помогите Кириллу составить какое-то количество команд так, чтобы i -й герой входил хотя бы в a_i команд, и суммарная сила этих команд была как можно меньше.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 10^5$) — количество героев.

Вторая строка содержит n целых чисел a_1, \dots, a_n ($1 \leq a_i \leq 10^9$) — количество команд, в которые должен входить каждый из героев.

Третья строка содержит n целых чисел p_1, \dots, p_n ($1 \leq p_i \leq 10^9$) — силы героев.

Формат выходных данных

Выведите единственное целое число — минимальную суммарную силу команд.

Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5 5 4 3 2 1	15
5 10 9 10 2 4 3 4 3 10000 1	20034

Замечание

В первом примере Кириллу выгодно составить 5 команд из героев с номерами из следующих отрезков: $[1, 5]$, $[2, 5]$, $[3, 5]$, $[4, 5]$ и $[5, 5]$. Силы этих команд равны 5, 4, 3, 2 и 1 соответственно, поэтому их суммарная сила равна $5 + 4 + 3 + 2 + 1 = 15$.

[А] 3. Секта фанатиков языка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мегабайта

Как известно, секты — страшные формирования, которые полностью меняют мировоззрение человека и изолируют его от общества. Секты не обошли стороной и программистов, и вот уже несколько лет среди программистов орудует страшная секта фанатиков языка программирования Rust.

Страна Берляндия долгое время была обделена вниманием секты, но разведка страны выяснила, что секта планирует выслать в Берляндию проповедников, которые начнут распространять опасный язык среди программистов страны.

Сама Берляндия состоит из n городов и m двухсторонних дорог между ними. i -я дорога соединяет города с номерами u_i и v_i и имеет длину l_i километров. Известно, что секта планирует высадить проповедников не более чем в k городах. У каждого проповедника есть своя величина x_i — сила убеждения, что Rust это лучший язык программирования и все надо переписать на него. Известно, что город a попадет под влияние секты, если существует хотя-бы один проповедник с силой убеждения x_i находящийся в городе b таком что существует путь от города b до города a по дорогам, у которых длины не превосходят x_i .

Правительство Берляндии не знает ни сил убеждений проповедников, ни номера городов, в которых они планируют высадиться. Берляндия хочет подготовиться ко всем возможным сценариям нашествия языка Rust. Поэтому вам поручена важная задача — посчитать количество различных вариантов множеств городов, которые окажутся под влиянием секты при всех возможных вариантах сил убеждения и начальных городов проповедников языка. Обратите внимание, что в Берляндию приедет не более k проповедников (возможно и 0). Два множества городов под влиянием секты считаются различными, если в одном множестве некоторый город находится под влиянием секты, а в другом — нет. Помогите спасти жителей Берляндии от страшного языка программирования Rust!

Формат входных данных

В первой строке даны три целых числа n , m и k ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$, $1 \leq k \leq 500$) — число городов и дорог в Берляндии, а также верхнее ограничение на количество приехавших проповедников.

В следующих m строках описываются дороги Берляндии. В i -й строке дано по три числа u_i , v_i и l_i ($1 \leq u_i, v_i \leq n$, $1 \leq l_i \leq 10^9$) — номера городов, между которыми проложена i -я дорога и ее длина.

Формат выходных данных

В единственной строке выведите одно число — количество различных возможных множеств городов, оказавшихся под влиянием секты. Так как это число может быть большим, выведите его по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 2 1 1 2 1 1 3 2	6
6 9 2 1 2 1 1 3 2 2 3 2 2 4 4 4 6 1 3 5 7 4 3 5 4 5 3 4 6 8	40
4 0 5	16

Замечание

В первом примере возможны следующие множества городов, попавшие под влияние секты: {}, {1}, {2}, {3}, {1, 2}, {1, 2, 3}. Множество {2, 3} невозможно, так как если секта смогла распространиться по дороге между городами 2 и 3, то она сможет распространиться и по более короткой дороге между городами 1 и 2.

[А] 4. Скуфоленд

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

Скуфоленд — это загадочный город с необычными жителями.

В Скуфоленде есть n районов, соединенных дорогами в виде дерева. Более формально, если принять районы за вершины, а дороги за ребра, то получится связный неориентированный граф с n вершинами и $n - 1$ ребрами. У каждой дороги есть длина, равная 0 или 1.

В этом городе живут m скуфов, i -й живет в районе u_i . Каждый из них уже выбрал альтушку, альтушка i -го скуфа живет в районе v_i . Но доставка в Скуфоленде не любит четные числа, поэтому i -й скуф получит свою девушку, только если из вершины u_i в вершину v_i существует путь нечетной длины (длина пути равна сумме длин ребер).

Мало всего этого, в Скуфоленде есть еще и система социального рейтинга. В ней все скуфы отсортированы по порядку, от 1-го до m -го, при этом у 1-го самый низкий рейтинг, а у m -го самый высокий. Правительство считает, что если рейтинг скуфа a меньше, чем скуфа b , и a получил альтушку, то b тоже должен получить свою. Соблюдение этого условия является делом государственной важности, однако единственный способ повлиять на это — выбрать длины всех дорог в городе (изначально длины дорог не фиксированы и правительство может назначить каждой длину, равную 0 или 1 по своему усмотрению).

Помогите правительству Скуфоленда и найдите количество способов назначить длины дорог так, чтобы условие выполнялось. Так как ответ может быть большим, выведите его по модулю 998 244 353.

Формат входных данных

В первой строке вводятся два числа n и m ($2 \leq n, m \leq 250000$) — число районов и число скуфов соответственно.

Во второй строке вводятся $n - 1$ число p_1, p_2, \dots, p_{n-1} ($1 \leq p_i \leq i$). i -е число означает, что есть дорога между районами p_i и $i + 1$.

В следующих m строках вводятся по два числа: u_i и v_i ($1 \leq u_i, v_i \leq n$) — районы где живут i -й скуф и его альтушка соответственно.

Формат выходных данных

Выведите одно целое число: число способов назначить длины дорог так, что бы условие выполнялось, по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
5 5 1 2 3 2 1 2 1 3 2 3 1 5 1 4	4
3 2 1 2 1 2 2 3	3
5 3 1 2 1 4 2 3 1 4 1 5	8

[А] 5. Скуфогонки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Теперь, когда у каждого скуфа есть альтушка, они решили устроить гонки!

Всего есть n скуфов, пронумерованных от 1 до n . Они находятся на прямой, на которой введены координаты. Изначально i -й скуф находится на координате a_i . Все a_i различны. В момент времени 0 все скуфы начинают бежать, при этом i -й скуф бежит со скоростью i **влево**. Это значит, что координата i -го скуфа уменьшается на i каждую секунду.

Так как у скуфов теперь есть подружки, то они тоже решили участвовать в гонках. Каждая альтушка забралась на плечи своего скуфа и может передвигаться на нем. Но альтушки не выбрали скуфов, поэтому они хотят уйти от своего скуфа и попасть к другому! Альтушка, изначально попавшая к i -му скуфу, на самом деле хочет оказаться с q_i -м скуфом! Для того, что бы достичь этого, альтушка может перепрыгнуть с одного скуфа на другого в тот момент, когда их координаты совпадают (этот момент не обязан быть целым). При этом возможны ситуации, когда на одном скуфе нет ни одной альтушки, или наоборот у одного скуфа есть сразу несколько альтушек!

Альтушки просят вас помочь им, и для каждой из них узнать, за какой минимальное время она может попасть к нужному скуфу.

Формат входных данных

В первой строке записано целое число n ($2 \leq n \leq 3 \cdot 10^5$) — число скуфов.

Во второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — начальные позиции скуфов.

В третьей строке записаны n целых чисел q_1, q_2, \dots, q_n ($1 \leq q_i \leq n, q_i \neq i$) — номера желаемых скуфов для каждой альтушки.

Гарантируется, что все a_i различны, и что $q_i \neq i$.

Формат выходных данных

Выведите n строк: для каждой альтушки минимальное время, когда она может попасть к желаемому скуфу, или -1 , если попасть к нему невозможно. Время следует выводить в виде несократимой дроби a/b , где a — положительно и a взаимно просто с b .

Примеры

стандартный ввод	стандартный вывод
3 2 5 11 3 3 1	9/2 9/2 9/2
4 1 12 10 3 4 3 2 3	2/3 -1 11/1 9/2
7 2 6 7 9 12 16 1 7 1 1 1 1 1 4	-1 7/3 7/3 7/3 5/2 14/5 -1

[А] 6. Шаи-Хулуд. Королевская битва

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

На планете Арракис обитают огромные песчаные черви, именуемые Шаи-Хулуд. Они могут перемещаться по планете, по городам и дорогам, их соединяющим. Дорожная инфраструктура планеты очень специфична. Города связаны дорогами таким образом, что никакие две дороги не соединяют одну и ту же пару городов, никакая дорога не ведёт из города в себя же, а также каждая дорога лежит не более чем на семи простых циклах.

Всего на Арракисе есть n городов и m дорог, при этом i -я дорога соединяет города с номерами u_i и v_i и имеет длину c_i километров. Так как система дорог на планете Арракис очень сложна, то черви решили ее улучшить, первым этапом улучшения является сбор статистики. В частности, Шаи-Хулуд хочет узнать сумму кратчайших расстояний между каждой парой городов. С этим запросом они отправились к вам.

К сожалению, вы долго игнорировали запрос Шаи-Хулуда, а когда вы решили приступить к задаче, то песчаная буря разрушила всю систему дорог планеты. После этих событий Шаи-Хулуд разгневался и прислал человечеству особое требование.

Шаи-Хулуд потребовал, чтобы 8-классники (особенно межнарники по информатике) вели себя более уважительно по отношению к другим участникам в чате Яндекс Кружка. К сожалению, это требование было проигнорировано, что стало роковой ошибкой для человечества. Шаи-Хулуд разгневался ещё сильнее и дал последнее задание человечеству, за невыполнение которого все человечество будет уничтожено. Именно вы должны выполнить это задание и спасти человечество!

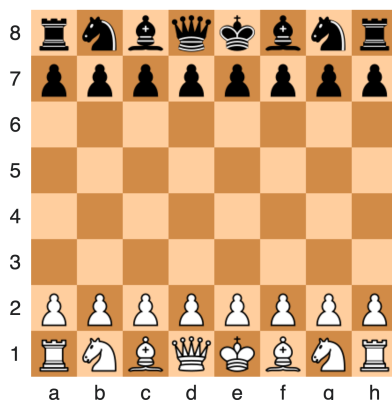
Как вы знаете по опыту прошлого года, Шаи-Хулуд очень любит шахматы. Обычно черви играют в шахматы на бесконечной шахматной доске с большим количеством фигур, однако в этот раз черви решили сделать для вас исключение и сыграть с вами в классические шахматы по стандартным правилам.

Правила шахмат

Эта секция почти полностью повторяет правила шахмат с Википедии с небольшими отклонениями. Отклонения будут отдельно указаны в данных правилах.

В шахматы играют на доске, поделённой на равные квадратные клетки, или поля. Размер доски — 8×8 клеток. Вертикальные ряды полей (вертикали) обозначаются латинскими буквами от a до h слева направо, горизонтальные ряды (горизонтали) — цифрами от 1 до 8 снизу вверх; каждое поле обозначается сочетанием соответствующих буквы и цифры. Поля раскрашены в тёмный и светлый цвета (и называются, соответственно, чёрными и белыми) так, что соседние по вертикали и горизонтали поля раскрашены в разные цвета. Доска располагается так, чтобы ближнее угловое поле справа от игрока было белым (для белых это поле $h1$, для чёрных — поле $a8$).

У игроков в начале игры имеется по одинаковому набору фигур. Фигуры одного из игроков условно называются «белыми», другого — «чёрными». Белые фигуры окрашены в светлый цвет, чёрные — в тёмный. Сами игроки называются «белые» и «чёрные» по цвету своих фигур.



В каждый комплект фигур входят: король, ферзь, две ладьи, два слона, два коня и восемь пешек. В начальной позиции фигуры обеих сторон размещаются так, как показано на диаграмме выше. Белые занимают первую и вторую горизонтали, чёрные — седьмую и восьмую. Пешки расположены на второй и седьмой горизонталях соответственно.

Игра заключается в том, что игроки поочерёдно делают ходы. Первый ход делают белые. За исключением *особых ходов* (взятие на проходе и рокировка, описание данных ходов доступно ниже), ход заключается в том, что игрок перемещает одну из своих фигур на другое поле по следующим правилам:

- Фигуры (кроме коня) передвигаются по прямой линии, при этом все промежуточные поля между начальным и конечным должны быть свободны (на них не должно быть своих или чужих фигур). Исключением является ход коня.
- Ход на поле, занятое своей фигурой, невозможен.
- При ходе на поле, занятое чужой фигурой, она снимается с доски (такой ход называется взятием).
- Король ходит на соседнюю клетку по вертикали, горизонтали или диагонали.
- Ферзь ходит на любое расстояние по вертикали, горизонтали или диагонали.
- Ладья ходит на любое расстояние по вертикали или горизонтали.
- Слон ходит на любое расстояние по диагонали.
- Конь двигается на две клетки по вертикали и затем на одну клетку по горизонтали, или наоборот, на две клетки по горизонтали и на одну клетку по вертикали, тем самым движение коня напоминает заглавную букву «Г» кириллического или заглавную букву «L» латинского алфавитов. Другими словами, конь ходит на одно из полей, ближайших к тому, на котором он стоит, но не на той же самой горизонтали, вертикали или диагонали. При этом цвет поля, на которое ходит конь, всегда противоположен цвету поля, на котором он стоит. Конь отличается от остальных фигур также тем, что другие фигуры, стоящие на пути коня, не препятствуют его ходу (он «перепрыгивает» через них).
- Пешка может ходить только вперёд (направлением «вперёд» называется направление к восьмой горизонтали для белых или к первой для чёрных): без взятия — на одно поле вперёд по вертикали, а со взятием — по диагонали на одно поле вперёд-вправо или вперёд-влево. Если пешка находится на начальном поле (вторая горизонталь для белых и седьмая для чёрных), то она также может сделать ход без взятия на два поля вперёд. Когда пешка ходит на последнюю горизонталь (для белых — на восьмую, для чёрных — на первую), она заменяется по выбору игрока на любую другую фигуру того же цвета, кроме короля и пешки (то есть пешка может превратиться в ферзя, ладью, слона или коня). Такой ход называется *превращением пешки*. Превращение пешки является частью того хода, которым она перемещается на последнюю горизонталь. Таким образом, если, например, превращённая из пешки фигура угрожает королю противника, то этот король немедленно оказывается под шахом. Нет никаких дополнительных ограничений на фигуру, в которую пешка может превратиться в результате превращения (то есть например на доске может стать больше одного ферзя).

Поле называется *битым*, если при своём ходе фигура могла бы взять находящуюся на этом поле фигуру противника (независимо от того, есть ли такая фигура на этом поле). Поле считается битым, даже если фактически ход фигурой туда невозможен, так как ставит под бой собственного короля.

Дополнительно существует два вида *особых ходов*:

- *Рокировка* — если король и одна из ладей того же цвета не двигались с начала игры, то король и эта ладья могут в один ход одновременно сменить положение (рокироваться). При рокировке король сдвигается на две клетки по направлению к ладье, а ладья ставится на

поле между начальной и конечной позицией короля. Рокировка невозможна, если король или соответствующая ладья уже ходили. Рокировка временно невозможна, если между королём и ладьёй находится какая-либо фигура, а также если поле, на котором стоит король, или поле, которое он должен пересечь, или поле, которое он должен занять, находится под ударом одной из фигур противника. Обратите внимание, что при шахе король не может сделать рокировку, даже если это перенесет его на небитое поле. Также обратите внимание, что поле, на котором исходно стояла ладья или поле, через которое пройдет ладья но на которое не встанет король может быть битым.

- *Взятие на проходе* — когда пешка совершает ход на две клетки через битое поле, находящееся под ударом пешки противника, то ответным ходом она может быть взята этой пешкой противника. При этом пешка противника перемещается на битое поле, а взятая пешка снимается с доски (пример показан на диаграмме). Взятие на проходе возможно только пешкой непосредственно в ответ на ход пешки через битое поле, на следующих ходах оно уже не разрешено.

Король, находящийся на битом поле, называется «*стоящим под шахом*». Сделать ход, после которого король противника оказывается под шахом, значит «*объявить шах*». **Ходы, после которых король сделавшего ход остаётся или оказывается под шахом, запрещены.** Игрок, король которого находится под шахом, обязан немедленно его устранить.

Если король игрока находится под шахом и игрок не имеет ни одного хода, позволяющего устранить этот шах, этот игрок называется *получившим мат* и, соответственно, он терпит поражение. Цель игры и состоит в том, чтобы поставить мат королю противника.

Если игрок при своей очереди хода не имеет возможности сделать ни одного хода по правилам, но король игрока не находится под шахом, такая ситуация называется «*пат*».

Игра завершается выигрышем одной из сторон или ничьей. Ничья фиксируется в следующих случаях (**правила ничьи ниже незначительно отличаются от классических правил шахмат, прочтите их внимательно**):

- Поставлен *пат*.
- Троекратное повторение позиции (не обязательно в течение трёх ходов подряд), причём в понятие позиции в этом случае входит только расположение фигур (данное определение отличается от классических правил шахмат).
- *Правило 50 ходов* — в течение 100 последних полуходов (ходов одного из игроков) не было сделано ни одного взятия и ни одного хода пешкой.

Мат имеет приоритет перед ничьей, то есть если после хода игрока второй игрок находится в матовом положении (но не патовом), то игра завершается победой игрока, сделавшего ход, даже если какое-то из условий ничьи выполнено.

Шахматная нотация

В этой секции описана нотация, используемая для записи ходов. В данной задаче будет использована полная запись хода (автор условия не справился разобраться в сокращённой).

Вертикали обозначаются латинскими буквами от *a* до *h* от ферзевого фланга к королевскому, горизонтали — цифрами от 1 до 8 от белых фигур к чёрным.

Каждое поле шахматной доски обозначается двумя символами — буквой вертикали и цифрой горизонтали, к которым принадлежит это поле. Так, в стандартной начальной расстановке белый король стоит на поле, имеющем обозначение *e1*, а чёрный — на поле *e8*.

Фигуры обозначаются по следующим правилам:

- Король: *K*
- Ферзь: *Q*
- Ладья: *R*

- Конь: N
- Слон: B
- Пешка: нет буквенного обозначения.

Каждый ход обозначается определенной строкой, составленной по следующему принципу:

1. В начале записывается обозначение фигуры, сделавшей ход (в случае хода пешки ничего не записывается).
2. Следующие два символа обозначают поле, с которого фигура сделала ход.
3. Следующий символ обозначает действие хода. В случае совершения взятия, записывается символ «x», иначе записывается символ «-». *Взятие на проходе* также считается стандартным взятием и в случае взятия на проходе действие хода обозначается символом x, при этом никаких дополнительных особенностей в обозначение данного хода нет.
4. Следующие два символа обозначают поле, на которое сделан ход.
5. Если пешка совершила *превращение*, в конце указывается обозначение фигуры, в которую она превратилась.

Для рокировки существует специальное обозначение хода. Рокировка в короткую сторону (при которой король перешел на вертикаль g) обозначается «0-0», рокировка в длинную сторону обозначается «0-0-0» (при обозначении рокировки используются заглавные буквы 0).

Ход записывается в одну строку без лишних пробельных символов.

Итак, теперь вы знаете правила шахмат (и возможно как автор условия офигели от того, что существует вот это вот взятие на проходе), а значит можно вернуться к Шаи-Хулуду. Шаи-Хулуд собирается сыграть с вами партию в шахматы. Чтобы человечество не было уничтожено, вы должны поставить мат Шаи-Хулуду (даже ничья не считается успешным завершением игры). Вы ходите за белых и делаете первый ход. К счастью, Шаи-Хулуд не очень умный и поэтому на каждом ходу игры будет случайно выбирать произвольный ход из всех доступных. Победите Шаи-Хулуда в шахматы и спасите человечество от страшного песчаного червя!

Протокол взаимодействия

Это интерактивная задача.

Во время игры вы с Шаи-Хулудом будете поочередно делать ходы, так как вы играете за белых, то вы начинаете первыми. Перед каждым вашим ходом (в том числе первым), интерактор будет в отдельной строке выводить статус игры. Это может быть одна из 3 строк:

- **playing** — данная строка обозначает, что игра идет и вам надо совершить очередной ход.
- **mate** — данная строка обозначает, что после последнего хода Шаи-Хулуда вам был поставлен мат. В этом случае ваше решение будет считаться неверным. После считывания этой строки ваша программа должна сразу завершиться, больше ничего не считывая и не выводя.
- **draw** — данная строка обозначает, что после последнего хода Шаи-Хулуда игра закончилась ничьей. В этом случае ваше решение будет считаться неверным. После считывания этой строки ваша программа должна сразу завершиться, больше ничего не считывая и не выводя.

Далее, вы должны вывести ваш ход в полной шахматной нотации (описано выше). Обратите внимание, вы обязательно должны выводить фигуру, которая ходит (за исключением пешки), и в случае взятия обязательно обозначать действие (буквой x). То есть у любого возможного хода есть только одно обозначение в соответствии с условием и вы должны обязательно использовать его. Не забывайте сбрасывать буффер после каждого вашего вывода.

В ответ на ваш ход интерактор выведет в отдельной строке текущий статус игры. Он может обозначаться одной из четырех возможных строк:

- **playing** — данная строка обозначает, что игра продолжается и далее будет ход чёрных.
- **mate** — данная строка обозначает, что после вашего хода Шаи-Хулуда был поставлен мат. В этом случае считается, что вы победили и ваше решение прошло тест. Далее ваша программа должна сразу завершиться, больше ничего не считывая и не выводя.
- **draw** — данная строка обозначает, что после вашего хода игра закончилась ничьей. В этом случае ваше решение будет считаться неверным. После считывания этой строки ваша программа должна сразу завершиться, больше ничего не считывая и не выводя.
- **wrong** — данная строка обозначает, что вы совершили некорректный ход. В этом случае ваше решение будет считаться неверным. После считывания этой строки ваша программа должна сразу завершиться, больше ничего не считывая и не выводя.

Далее, в отдельной строке будет выведен ход Шаи-Хулуда за черных, используя полную шахматную нотацию (описано выше).

После этого взаимодействие повторяется.

Пример

стандартный ввод	стандартный вывод
playing	e2-e4
playing e7-e5	
playing	Bf1-c4
playing Nb8-c6	
playing	Qd1-h5
playing Ng8-f6	
playing	Qh5xf7
mate	

Замечание

В данной задаче 150 тестов, тесты отличаются начальным инициализатором рандома стратегии Шаи-Хулуда. Для каждого теста инициализатор рандома задан заранее и не меняется.

Пример показан для ознакомления со форматом взаимодействия, стратегия Шаи-Хулуда на первом примере может вести себя не так, как в примере.

Для удобства отладки вам будет доступен полный вердикт на первом тесте из условия, в случае неправильного ответа на первом тесте вам будет показано сообщение интерактора.

Интерактор был заранее протестирован на корректность, однако если вы считаете, что в интеракторе задачи есть ошибки (сложно такое без ошибок написать), напишите в telegram @grphil.