Задача A. Обмены символов

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

Две строки s и t назовем *похожими*, если можно переставить символы в s так, чтобы получилась строка t.

Даны две строки s и t длины n, состоящие из строчных латинских букв. Обозначим их символы $s_1 s_2 \dots s_n$ и $t_1 t_2 \dots t_n$, соответственно. Для каждой позиции $i \in \{1, 2, \dots, n\}$ можно сделать операцию *обмена* — поменять местами символы s_i и t_i.

Определите, можно ли при помощи операций обмена сделать строки s и t похожими, и, если это возможно, выведите необходимые операции.

Формат входных данных

Первая строка входного файла содержит число n — длину обеих строк ($1 \leqslant n \leqslant 2 \cdot 10^5$).

Следующие две строки содержат строки s и t, соответственно. Гарантируется, что каждая из них имеет длину n и состоит из строчных латинских букв.

Формат выходных данных

Если при помощи операций обмена невозможно сделать строки похожими, выведите одно число -1.

Если существует набор операций, после которого s и t становятся похожими, выведите в первой строке число k — количество операций ($0 \leqslant k \leqslant n$). Во второй строке выведите k **различных** чисел $i_1, \dots, i_k$ — номера позиций, с которыми осуществляется операция обмена.

Примеры

стандартный ввод	стандартный вывод
4 aabc bcdd	2 2 4
3 abc abd	-1
7 players parsley	0

Задача B. Прибавляй + Присваивай = 2D

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

После прослушивания лекции по массовым операциям в дереве отрезков Майк и Дастин очень захотели применить свои знания. Майку очень понравилась возможность прибавлять на отрезке, а Дастину — присваивать на отрезке. Для того чтобы их действия не мешали друг другу, они решили их выполнять не на массиве, а в матрице $n \times m$. Изначально все числа в матрице были нулевыми. Майк и Дастин выполнили q операций трёх типов:

- `1 l r x`. Майк прибавляет x ко всем числам, лежащим в столбцах $l, l+1, \ldots, r$ (то есть, ко всем клеткам с координатами (x, y), где $l \leqslant y \leqslant r$).
- `2 i x`. Дастин присваивает x всем числам, лежащим в строке i (то есть, всем клеткам с координатами (i, y)).
- `3 i j`. Майк и Дастин записывают значение элемента в клетке с координатами (i, j).

Естественно, Майк и Дастин логгировали свои действия, но все эти логи были зашифрованы. Полученные логи имеют следующий вид:

- `1 p q y` для операций первого типа;
- `2 s y` для операций второго типа;
- `3 s c` для операций третьего типа.

Для расшифровки логов нужно поддерживать результат последней операции третьего типа α (до первой такой операции считается $\alpha = 0$) и вычислять реальные параметры операций по следующим формулам:

$$x = 1 + \big((\alpha + y) \bmod 10^9\big)$$
$$i = 1 + \big((\alpha + s) \bmod n\big)$$
$$j = 1 + \big((\alpha + c) \bmod m\big)$$
$$l = 1 + \big((\alpha + p) \bmod m\big)$$
$$r = 1 + \big((\alpha + q) \bmod m\big)$$

Восстановите вычисления Майка и Дастина и выведите результаты всех операций третьего типа.

Формат входных данных

В первой строке указана тройка n, m и q ($1 \leqslant n, m, q \leqslant 200\,000$) — размеры таблицы и количество операций.

Затем следуют описания операций в следующем формате:

- `1 p q y`. ($1 \leqslant p, q \leqslant m$, $1 \leqslant y \leqslant 10^9$)
- `2 s y`. ($1 \leqslant i \leqslant n$, $1 \leqslant y \leqslant 10^9$)
- `3 s c`. ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$)

Гарантируется, что после расшифровки $l \leqslant r$ для всех операций первого типа.

Формат выходных данных

Для каждой операции третьего типа выведите ответ на него в отдельной строке.

Пример

стандартный ввод	стандартный вывод
3 3 9 1 3 1 1000000000 3 1 1 2 1 1000000000 3 1 1 3 3 1 1 2 3 1000000000 3 3 2 3 2 3 3 3 1	1 2 2 5 3 4

Задача C. Оптимизируем цензурирование

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

Вы служите младшим редактором в одном из отделов Министерства Правды. Ежедневно из глубин архивов поступают документы, созданные задолго до Великого Переписывания. В этих записях всё ещё встречаются выражения, признанные ныне *идеологически неверными*: они вступают в противоречие с принципами ангсоца, умаляют образ Старшего Брата или подразумевают наличие альтернативных взглядов. Такие слова запрещено не только произносить — о них не должно возникать даже *мысли*.

Однако древние тексты нельзя уничтожить полностью, поэтому их требуется **очищать**. Имеется строка S, в которой могут содержаться запрещённые слова $T_1, T_2, \ldots, T_Q$. Для приведения текста к нормам действующей редакции *Новояза* разрешено заменять отдельные символы специальным знаком «*». Каждая замена трактуется как проявление верности Партии, но чрезмерная активность способна привлечь внимание внутренней полиции, поэтому число замен должно быть минимальным.

Определите наименьшее количество символов, которые необходимо заменить на «*», чтобы ни одно из запрещённых слов не встречалось в результирующей строке. После правки ни одно слово T_i не должно присутствовать в тексте целиком.

Формат входных данных

В первой строке дана строка S, состоящая из строчных латинских букв ($1 \leqslant |S| \leqslant 5 \cdot 10^5$).

Во второй строке задано число Q — количество запрещённых слов ($1 \leqslant Q \leqslant 5 \cdot 10^5$). В последующих Q строках заданы строки T_i, каждая из которых состоит из строчных латинских букв ($1 \leqslant |T_i|$, а также $\sum_{i=1}^{Q} |T_i| \leqslant 5 \cdot 10^5$).

Формат выходных данных

Выведите одно целое число — минимальное количество замен символов на «*», необходимое для полного устранения всех вхождений запрещённых слов из строки S.

Примеры

стандартный ввод	стандартный вывод
ilovedemocracy 3 i love democracy	3
lovebigbrother 1 myself	0
emmanuelgoldstein 2 gold goldstein	1

Задача D. Поиск общего соседа

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

К графу на N вершинах, в котором нет рёбер, поступают Q запросов двух типов:

- `1 u v` — добавить в граф ребро, соединяющее вершины u и v;
- `2 u v` — определить номер вершины, которая соединена ребром и с u, и с v, или сообщить, что такой нет.

Гарантируется, что в графе **никогда** не появятся циклы. Таким образом, ответ на второй запрос определяется однозначно. Если общего соседа у вершин u и v не существует, то ответом на второй запрос считается число 0.

Запросы даны в зашифрованном виде. А именно, вместо явно заданных параметров запроса t, u, v (где $t \in \{1, 2\}$ — тип) Вам будут даны параметры a, b, c. Пусть X это ответ на предыдущий запрос второго типа ($X = 0$ для первого запроса). Тогда реальные параметры запроса могут быть вычислены по формулам

- $t = 1 + ((a \cdot (1 + X)) \bmod 998\,244\,353) \bmod 2$;
- $u = 1 + ((b \cdot (1 + X)) \bmod 998\,244\,353) \bmod N$;
- $v = 1 + ((c \cdot (1 + X)) \bmod 998\,244\,353) \bmod N$.

Формат входных данных

В первой строке входных данных вводятся два числа N и Q — количество вершин и запросов ($2 \leqslant N \leqslant 10^5, 1 \leqslant Q \leqslant 10^5$).

В каждой из следующих строк вводятся зашифрованные параметры запросов a, b, c ($0 \leqslant a, b, c < 998\,244\,353$). Гарантируется, что в расшифрованных запросах $u \neq v$, а также что в графе никогда не появятся циклы.

Формат выходных данных

На каждый запрос второго типа выведите одно число — 0, если нет общего соседа у u и v, и номер общего соседа, иначе.

Примеры

<table>
<tr><th>стандартный ввод</th><th>стандартный вывод</th></tr>
<tr><td><pre>6 12
143209915 123840720 97293110
89822758 207184717 689046893
67757230 270920641 26993265
952858464 221010240 871605818
730183361 147726243 445163345
963432357 295317852 195433903
120904472 106195318 615645575
817920568 27584394 770578609
38727673 250957656 506822697
139174867 566158852 412971999
205467538 606353836 855642999
159292205 319166257 51234344</pre></td><td><pre>0
2
0
2
6
0
1</pre></td></tr>
<tr><td><pre>2 1
377373366 41280240 33617925</pre></td><td></td></tr>
</table>

Замечание

В примере расшифрованные запросы это:

```
6 12
2 1 3
1 2 6
1 2 4
1 1 3
2 4 6
2 1 4
1 5 6
1 1 2
2 1 4
2 2 5
2 3 4
2 2 3
```

Задача E. Техники сбора букетов

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

Питер — любопытный и умный студент. У него назначена встреча с Гвен, и он хочет подарить ей восхитительный букет. Город, в котором они живут, представляет собой n перекрестков, соединенных $n-1$ дорогами. По дорогам можно добраться от каждого перекрестка до каждого. Очень удобно, что возле каждого перекрестка есть цветочный магазин. В цветочном магазине, находящемся у перекрестка с номером i, можно купить a_i различных корзин для цветов и b_i различных цветов. Цветы и корзины, продаваемые в разных магазинах, также различны.

Питер хочет собрать оригинальный букет, и для этого будет действовать следующим образом. Из перекрестка с номером s он пойдет к перекрестку с номером t по кратчайшему пути. В каком-то магазине, находящемся у перекрестка на пути его следования, он купит корзину. С имеющейся корзиной он будет заходить в магазин у каждого следующего перекрестка на пути и брать в них не более одного цветка. Наблюдательный Питер сразу понял, что у него есть очень много вариантов букетов, которые он может собрать (естественно, в хорошем букете точно должна быть и корзина, и хотя бы один цветок). Ему стало интересно узнать точное количество различных способов собрать букет.

Место встречи еще не определено точно, и есть m возможных вариантов маршрута Питера, в которых он начинает у перекрестка s_j и заканчивает у перекрестка t_j. Помогите Питеру посчитать количество возможных способов собрать букет для каждого варианта маршрута. Так как это число способов может быть слишком большим, от Вас требуется найти только его остаток от деления на $998\,244\,353$.

Формат входных данных

В первой строке входных данных вводится одно целое число n ($1 \leqslant n \leqslant 2 \cdot 10^5$) — количество перекрестков в городе.

В следующей строке вводятся n целых чисел $a_1 ... a_n$ ($0 \leqslant a_i < 998\,244\,353$) — количества различных корзин в магазине у i-го перекрестка. В следующей строке вводятся n целых чисел $b_1 ... b_n$ ($0 \leqslant b_i < 998\,244\,353$) — количества различных цветков в магазине у i-го перекрестка.

В каждой из следующих $n-1$ строк вводятся два целых числа u и v ($1 \leqslant u, v \leqslant n$, $u \neq v$) — номера перекрестков, которые соединены очередной дорогой.

В следующей строке вводится одно целое число m ($1 \leqslant m \leqslant 2 \cdot 10^5$) — количество вариантов маршрута Питера. В каждой из следующих m строк вводятся два целых числа s_j и t_j ($1 \leqslant s_j, t_j \leqslant n$) — начальный и конечный перекресток в очередном варианте маршрута.

Формат выходных данных

Выведите m целых чисел — для каждого варианта маршрута остаток от деления числа способов собрать букет на $998\,244\,353$.

Примеры

стандартный ввод	стандартный вывод
3 2 3 5 4 6 1 1 2 2 3 6 1 2 1 3 2 3 2 1 3 1 3 2	12 29 3 12 182 30
4 1 0 0 0 1 1 2 3 1 2 2 3 2 4 5 1 1 1 2 1 3 1 4 3 4	0 1 5 7 0
6 17 917 957 17 957 57 23 92 395 72 395 7 4 3 2 6 3 2 5 2 6 1 5 6 5 1 3 3 1 1 1 2 3	2461354 7469945 17263693 0 362215

Замечание

В первом примере в первом запросе Питер точно должен купить корзинку у перекрестка 1, а цветок у перекрестка 2. Поэтому всего есть $2 \cdot 6 = 12$ способов это сделать.

В первом примере во втором запросе, если Питер берет корзинку у перекрестка 1, то далее он либо берет по цветку у обоих следующих перекрестков ($6 \cdot 1$ способ), либо в каждом по одному (еще $6 + 1$ способ), если же Питер купит корзинку у перекрестка 2, то он должен будет купить цветок у перекрестка 3. Поэтому, итого $2 \cdot (6 \cdot 1 + 6 + 1) + 3 \cdot 1 = 29$ способов.

Задача F. Сравнение сдвигов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Для строки X обозначим X^i — циклический сдвиг строки X на i символов влево (то есть, если символы X это $x_1 x_2 \dots x_N$, то $X^i = x_{i+1} x_{i+2} \dots x_N x_1 x_2 \dots x_i$).

Даны две строки S и T длины N. От Вас требуется вычислить количество пар (i, j) таких, что $0 \leqslant i, j \leqslant N - 1$ и строка S^i лексикографически меньше или равна строке T^j.

Формат входных данных

В первой строке входных данных дано одно число N — длина строк ($1 \leqslant N \leqslant 2 \cdot 10^5$).

В следующих двух строках даны S и T, соответственно. Гарантируется, что они имеют длину N и состоят из строчных латинских букв.

Формат выходных данных

В единственной строке выведите одно число — количество пар (i, j) таких, что $S^i \leqslant T^j$.

Примеры

стандартный ввод	стандартный вывод

`3`
`aba`
`bca`
	`8`

`1`
`b`
`a`
	`0`

`14`
`tbankparallelx`
`orparallelcbpb`
	`91`

Замечание

Во втором примере все циклические сдвиги S это `aab`, `aba`, `baa`, все циклические сдвиги t это `abc`, `bca`, `cab`.

Задача G. Увеличиваем среднее арифметическое

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мебибайт

Начало января — самое время подводить итоги прошлого года. Мирон работает в статистическом бюро и готовит годовой отчет. Статистическое бюро проводит n измерений некоторого параметра за год. Результаты измерений заданы целыми неотрицательными числами $a_1, a_2, \ldots, a_n$.

Для отчёта выбирается *ключевое измерение* с номером i. После этого Мирон может выбрать любой последовательный отрезок $[l, r]$ так, чтобы в него входило ключевое измерение $(1 \leqslant l \leqslant i \leqslant r \leqslant n)$, и вписать в отчёт среднее значение на этом отрезке:

$$\frac{a_l + a_{l+1} + \cdots + a_r}{r - l + 1}.$$

Мирон выбирает отрезок так, чтобы среднее было как можно больше.

Помогите Мирону составить отчет. Для каждого возможного ключевого измерения i вычислите максимальное среднее арифметическое параметра по всем отрезкам, содержащим позицию i.

Формат входных данных

В первой строке дано единственное число n $(1 \leqslant n \leqslant 2 \cdot 10^5)$ — количество измерений.

Во второй строке даны n чисел $a_1, \ldots, a_n$ $(0 \leqslant a_i \leqslant 10^9)$ — измерения параметра.

Формат выходных данных

Выведите n чисел в отдельных строках.

Ответ в i-й строке должен быть равен максимальному среднему арифметическому по всем отрезкам, содержащих i-е измерение. Ответ считается верным, если абсолютная или относительная погрешность не превышает 10^{-6}

Пример

стандартный ввод	*стандартный вывод*
6 2 4 5 7 3 6	4.500000 5.333333 6.000000 7.000000 5.333333 6.000000

Задача H. Нулевой XOR в графе

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

Операция XOR (битовое исключающее ИЛИ) обладает очень интересным свойством: можно вычислить её от большого числа ненулевых чисел и получить ноль (например, $3 \oplus 2 \oplus 1 = 0$).

Лукас и Дастин изучали один очень интересный граф, когда Дастин заявил, что может расставить в вершинах графа ненулевые числа так, чтобы для **каждой** вершины XOR чисел, записанных в её соседях, был равен нулю. Лукас не поверил в утверждение Дастина, и у них начался спор.

Помогите Дастину найти подходящую расстановку чисел (причём они не должны быть слишком большими — их значения должны быть до $2^{60} - 1$ включительно) или сообщите, что он проиграл спор.

Формат входных данных

В первой строке входных данных вводятся два целых числа n и m — количество вершин и рёбер в графе ($1 \leqslant n \leqslant 60$, $0 \leqslant m \leqslant \frac{n(n-1)}{2}$). В каждой из следующих m строк вводятся u_i и v_i — номера вершин, соединённых i-м ребром ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$). Гарантируется, что все рёбра различны.

Формат выходных данных

Если Лукас выиграл в споре и искомого способа расставить числа не существует, выведите в единственной строке выходных данных слово «No».

Иначе, выведите в первой строке слово «Yes». В следующей строке выведите n чисел $c_1, c_2, \ldots, c_n$ — числа, записанные в вершинах для выполнения условия Дастина ($1 \leqslant c_i \leqslant 2^{60} - 1$).

Примеры

стандартный ввод	стандартный вывод
2 0	Yes 1152921504606846975 1
4 4 1 2 2 3 3 4 4 1	Yes 179 239 179 239
6 7 1 2 2 3 3 4 4 1 2 5 4 6 5 6	Yes 3 1 2 1 1 1
2 1 1 2	No

Замечание

XOR пустого множества полагается равным нулю, поэтому, если у вершины нет соседей, условие для неё автоматически выполнено.

Задача I. Сумма длин максимумов

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

Дан массив $A_1 \ldots A_N$, состоящий из натуральных чисел. *Ценой* подотрезка этого массива $A_l, A_{l+1}, \ldots, A_r$ называется величина $\max(A_l, A_{l+1}, \ldots, A_r) \cdot (r - l + 1)$. Требуется посчитать **сумму цен подотрезков массива по всем возможным разбиениям** массива на подотрезки. Так как эта величина может быть слишком большой, от вас требуется вывести только её остаток при делении на $998\,244\,353$.

Формат входных данных

В первой строке входных данных вводится одно число N $(1 \leqslant N \leqslant 10^6)$.

В следующей строке вводятся N целых чисел $A_1 ... A_N$ $(1 \leqslant A_i < 998\,244\,353)$.

Формат выходных данных

Выведите единственное число — остаток от деления искомой суммы на $998\,244\,353$.

Примеры

стандартный ввод	стандартный вывод
3 1 3 2	30
5 3 1 1 2 4	236
5 89981 67859 83383 46618 56031	6228390

Замечание

В первом примере массив $[1, 3, 2]$ можно разбить на подотрезки 4 способами:

1. $[1], [3], [2]$;
2. $[1], [3, 2]$;
3. $[1, 3], [2]$;
4. $[1, 3, 2]$.

Поэтому ответ на задачу это $(1 \cdot 1 + 3 \cdot 1 + 2 \cdot 1) + (1 \cdot 1 + 3 \cdot 2) + (3 \cdot 2 + 2 \cdot 1) + (3 \cdot 3) = 6 + 7 + 8 + 9 = 30$.

Задача J. НОД второго порядка

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

Все знают, что наибольший общий делитель — очень полезная функция. Однажды в ходе математических изысканий Саша придумал новую функцию — *наибольший общий делитель второго порядка*. Она определяется для набора из $k \geqslant 2$ чисел следующим образом:

$$\gcd_2(b_1, b_2, \ldots, b_k) = \gcd(b_1 \cdot b_2, b_1 \cdot b_3, \ldots, b_i \cdot b_j, \ldots, b_{k-1} \cdot b_k) = \gcd(b_i \cdot b_j | 1 \leqslant i < j \leqslant k)$$

Иными словами, $\gcd_2$ для набора чисел $(b_1, \ldots, b_k)$ это НОД попарных произведений $b_i \cdot b_j$ по всем парам $i < j$. Для одного числа наибольший общий делитель второго порядка полагается равным нулю

$$\gcd_2(b_1) = 0.$$

Дан массив натуральных чисел $a_1, \ldots, a_n$. От Вас требуется вычислить сумму наибольших общих делителей второго порядка по всем подотрезкам массива. Поскольку эта сумма может быть очень большой, вычислите её остаток по модулю $10^9 + 7$. Формально, от Вас требуется вычислить следующую сумму:

$$\left(\sum_{1 \leqslant l \leqslant r \leqslant n} \gcd_2(a_l, a_{l+1}, \ldots, a_r) \right) \bmod 10^9 + 7.$$

Формат входных данных

В первой строке дано число t ($1 \leqslant t \leqslant 20\,000$) — количество наборов входных данных.

Каждый набор входных данных описывается парой строк. В первой строке содержится n ($2 \leqslant n \leqslant 10^5$) — длина массива a, а во второй содержатся числа $a_1, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^{18}$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит $200\,000$.

Формат выходных данных

Для каждого набора входных данных выведите единственное число x ($0 \leqslant x \leqslant 10^9 + 6$) — значение искомой суммы.

Пример

стандартный ввод	стандартный вывод
4 3 1 2 3 3 2 2 2 6 8 4 2 6 3 9 4 1 3 3 1	9 12 137 22

Замечание

- $\gcd_2(1) + \gcd_2(1,2) + \gcd_2(1,2,3) + \gcd_2(2) + \gcd_2(2,3) + \gcd_2(3) = 0 + 2 + \gcd(2,3,6) + 0 + 6 + 0 = 2 + 1 + 6 = 9$.
- $\gcd_2(2) + \gcd_2(2,2) + \gcd_2(2,2,2) + \gcd_2(2) + \gcd_2(2,2) + \gcd_2(2) = 0 + 4 + 4 + 0 + 4 + 0 = 12$.
- В четвертом тестовом примере на отрезке $[1, 4]$ значение $\gcd_2$ равно 1, на отрезке $[2, 3]$ значение $\gcd_2$ равно 9, на отрезках длины 1 значение $\gcd_2$ равно 0, а на остальных четырех отрезках равно 3. Значит, ответ равен $1 + 9 + 4 \cdot 3 = 22$.