

[C] 1. Красивые коктейли

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

Серёжа — профессиональный бармен. Он мешает коктейли. Сейчас в коктейле N разноцветных слоёв, каждый слой обозначен некоторой строчной латинской буквой (одинаковые слои — одинаковыми буквами, разные слои — разными буквами).

Серёжа хочет сделать коктейль *красивым*. У него весьма своеобразное понятие красоты: коктейль Серёжа считает *красивым*, если он весь одного цвета. Чтобы этого добиться, Серёжа может делать *перемешивания*: за одно *перемешивание* каждый слой, кроме последнего, либо остаётся такого же цвета, либо меняет цвет на цвет следующего слоя; а самый последний слой расплёскивается и исчезает.

Серёжа не хочет разливать слишком уж много коктейля. Помогите ему: скажите, за какое минимальное число *перемешиваний* Серёжа может сделать коктейль *красивым*.

Формат входных данных

На вход подаётся одна строка s длины $N \leq 100$, состоящая из строчных латинских букв.

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
colada	3
margarita	3
aaaaaaaaaa	0

Замечание

В первом примере Серёжа может выполнить, например, следующую серию перемешиваний:
colada → ollad → olld → ll

Во втором примере Серёжа может выполнить, например, следующую серию перемешиваний:
margarita → mrrgrtt → mrrgrt → rrrrr

В третьем примере коктейль уже изначально красивый, и ничего перемешивать не надо.

[C] 2. Напитки из Черноголовки

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

Никита очень любит лимонад из Черноголовки. Во время перерыва в занятии он решил купить немного своего любимого напитка. Никита знает, что в магазине лимонад продаётся в бутылках по 0.25, 0.5, 1 и 2 литра, которые стоят a, b, c, d рублей соответственно. Никита хочет купить **ровно** n литров лимонада, потратив при этом как можно меньше денег. Во сколько рублей Никите обойдётся покупка?

Формат входных данных

В первой строке входных данных вводится четыре целых положительных числа $a, b, c, d \leq 10^8$ — стоимости бутылок лимонада по 0.25, 0.5, 1, 2 литра соответственно.

Во второй строке входных данных вводится единственное целое положительное число $n \leq 10^9$ — количество литров лимонада, которое хочет купить Никита.

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
18 30 70 100 3	160
9876 987 98 9 1	98

Замечание

В первом примере выгодно купить одну двухлитровую бутылку и две бутылки по поллитра. Тогда Никита потратит $100 + 30 + 30 = 160$ рублей.

Во втором примере, хотя двухлитровая бутылка стоит дешевле литровой, Никита хочет купить **ровно** 1 литр лимонада, поэтому ему придётся потратить 98 рублей.

[C] 3. Грязные слова

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

У Егора есть хобби – он любит делать слова в произвольных текстах *грязными*. Чтобы сделать из некоторого слова грязное, надо оставить на своих местах первую и последнюю буквы, а затем перемешать все остальные буквы, слово при этом должно измениться. Если же сделать слово грязным не получается (как ни перемешивай – слово не меняется), то его оставляют нетронутым.

Ваша задача – помочь Егору сделать как можно больше слов текста грязными. Помогите ему и преобразуйте строку нужным образом!

Формат входных данных

На вход подается текст, состоящий из маленьких и больших латинских букв, знаков «!?:;-.» и пробелов. Словом называется последовательность подряд идущих маленьких и больших латинских букв, которую нельзя расширить влево или вправо.

Формат выходных данных

Выведите преобразованный текст, в котором как можно больше слов превращены в грязные.

Вывод должен кончаться переносом строки.

Если слово можно сделать грязным несколькими способами, то подойдет любой.

Примеры

стандартный ввод	стандартный вывод
I want--to!!!!break??Free	I wnat--to!!!!berak??Fere
Cat is not bread	Cat is not berad

[C] 4. Ёж в казино

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

Пытаясь найти нормальный курс по c++, Ёж набрел на детское онлайн-казино. Так как у Ежа есть большой опыт участия в различных махинациях, то ему не составило труда разгадать выигрышную стратегию.

В детском онлайн-казино можно нажимать на три разные клавиши («q», «w», «e»), пока детские алгоритмы вычисляют ваш выигрыш. Если нажимать на них в нужной последовательности, то шансы на выигрыш стремятся к 100 процентам!

Порядок нажатия клавиш таков:

- В течение a секунд надо держать зажатой клавишу «q»
- В течение b секунд надо быстро-быстро нажимать на клавишу «q» (при этом общее время, когда клавиша была зажата равно половине времени, когда на нее интенсивно тыкали). При этом если на клавишу «q» тыкали t секунд, после чего надо было покинуть казино, то при делении на 2 результат будет округляться вверх
- В течение c секунд надо держать зажатой клавишу «w»
- В течение d секунд надо держать зажатой клавишу «e»
- В течение e секунд надо держать одновременно зажатыми клавиши «w» и «e»

Затем Ёж повторяет все шаги с первого, чтобы сделать свой выигрыш ещё больше.

Суммарно Ёж собирается провести в казино K секунд. Посчитайте, сколько времени он будет держать зажатой каждую из клавиш.

Формат входных данных

Первая строка содержит пять целых чисел a, b, c, d, e ($1 \leq a, b, c, d, e \leq 100$), b — четно. Вторая строка содержит целое число K ($1 \leq K \leq 10000$) — сколько секунд Ёж проведёт в казино.

Формат выходных данных

Выведите в одну строку через пробел три числа — сколько секунд Ёж будет нажимать на клавиши «e», «w» и «q» соответственно.

Примеры

стандартный ввод	стандартный вывод
2 2 2 2 2 7	1 2 3
2 2 2 2 2 20	8 8 6
5 4 1 1 1 8	0 0 7

Замечание

В первом примере Ёж сначала будет зажимать 2 секунды клавишу «q», затем 2 секунды будет тыкать на «q», суммарно на «q» он жал 3 секунды. После этого у него останется 3 секунды на последующие нажатия. 2 секунды он будет нажимать на «w» и 1 секунду на «e», после чего время кончится.

Во втором примере Ёж успеет совершить два полных прохода по пунктам 1-5, на каждом из которых на «e» будет нажимать 4 секунды, на «w» тоже 4 секунды, а на «q» 3 секунды.

В третьем примере Ёж будет зажимать «q» 5 секунд, после чего у него останется 3 секунды, чтобы интенсивно тыкать на «q», то есть в ответ от этого времени прибавятся 2 секунды.

[C] 5. Полина и массив

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

Полина любит разнообразие, поэтому на день рождения друзья подарили ей массив целых чисел длины n . К сожалению, массив мог оказаться недостаточно разнообразным для Полины, поэтому она захотела поменять его так, чтобы:

- для всех i от 1 до n сумма первых i чисел массива не равнялась 0;
- для всех i от 2 до n сумма первых i чисел массива имела знак противоположный знаку суммы первых $i - 1$ чисел.

Чтобы не расстраивать друзей, Полина решила менять массив незаметно, а именно за одно изменение она может изменить ровно одно число не более чем на 1. Так, из числа 1 за одну операцию можно получить 2 или 0, а из числа -2 можно получить -3 или -1.

Помогите Полине! Посчитайте, за какое минимальное число изменений можно получить разнообразный массив.

Формат входных данных

В первой строке вводится число n ($1 \leq n \leq 10^5$). Во второй строке вводится n целых чисел, по модулю не превосходящих 10^9 .

Формат выходных данных

Выведите одно целое число — минимальное число изменений для получения интересного массива.

Примеры

стандартный ввод	стандартный вывод
3 1 -2 1	1
4 4 -7 5 -6	0

Замечание

В первом примере сумма первых трех чисел равна 0, но можно сделать массив разнообразным, увеличив последнюю 1. Получится массив 1 -2 2, который удовлетворяет описанным условиям.

Во втором примере ничего делать не нужно, так как массив и так удовлетворяет условиям.

[C] 6. Раздел торта

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

Алисе подарили прямоугольный торт размером n на m сантиметров. К сожалению, у Алисы есть два друга: Егор и Полина. А с друзьями, как всем известно, принято делиться.

Торт решили разрезать на три прямоугольных куска, причем:

- длина каждой стороны должна измеряться целым положительным числом сантиметров;
- сумма периметров кусков торта должна быть равна c

Перед поиском решения друзьям стало интересно: а можно ли так вообще сделать? Помогите им!

Формат входных данных

В единственной строке записаны три числа разделенные пробелом: n , m и c ($1 \leq n, m, c \leq 10^5$).

Формат выходных данных

Выведите «YES», если торт можно разрезать, выполнив условия выше. В противном случае выведите «NO».

Примеры

стандартный ввод	стандартный вывод
2 2 14	YES
2 2 13	NO

[C-B'] 1. Подготовка олимпиады

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

Прямо сейчас жюри одного небезызвестного лагеря готовит констест. У них есть n задач, которые теперь надо отсортировать по возрастанию. Эту задачу поручили `dshindov567889`.

Каждая задача имеет свой собственный номер — число от 1 до n . Более того, все номера задач различны. Но есть небольшой нюанс — сейчас они расположены в случайном порядке. За одну минуту наш герой может взять первую задачу из списка и переместить её в любое место в констесте. Однако `dshindov567889` любит экономить время, поэтому хочет сделать как можно меньше таких операций. Например, если изначальный тур выглядел следующим образом: $\{5, 4, 2, 1, 3\}$, то подойдет следующая последовательность операций:

1. Переместить задачу 5 в конец констеста. После этого констест будет выглядеть так: $\{4, 2, 1, 3, 5\}$
2. Переместить задачу 4 на предпоследнее место. Теперь констест такой: $\{2, 1, 3, 4, 5\}$
3. Переместить задачу 2 на вторую позицию. После этого констест будет отсортирован: $\{1, 2, 3, 4, 5\}$

Таким образом ответ на этот тест равен трем.

Помогите бедному `dshindov567889`, а после этого, может быть, он поможет и вам.

Формат входных данных

В первой строке находится число n ($1 \leq n \leq 3 \cdot 10^5$).

В следующей строке вводятся n различных чисел, разделенные пробелами. Каждое число находится в промежутке от 1 до n включительно.

Формат выходных данных

Выведите единственное число — минимальное количество операций, которое потребуется `dshindov567889` для того, чтобы сделать констест отсортированным по возрастанию.

Примеры

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

[C-B'] 2. Слежка за детьми

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

В Липецке недавно открылась Липецкая Компьютерная Школа (сокращенно — ЛКШ). Рассмотрим один из компьютерных классов в этом прекрасном месте.

Класс можно представить как бесконечно длинную прямую. В некоторых ее точках сидят группы учеников, которые могут между собой переговариваться, срывая тем самым контекст. Для простоты будем считать, что всего есть n таких точек: $(x_1, c_1); \dots; (x_n, c_n)$, означающие, что в точке x_i находятся c_i учеников. Но школа выделила специально обученного человека (наблюдателя), который может одновременно смотреть в две стороны!

Пусть наблюдатель стоит в точке p и имеет радиус зрения k . Тогда он видит всех учеников на отрезке $[p-k; p+k]$. Теперь наблюдатель хочет встать так, чтобы видеть как можно больше учеников. Вы — молодой стажер, поэтому Вам выпала задача посчитать, какое максимальное количество школьников одновременно может увидеть наблюдатель.

Формат входных данных

В первой через пробел вводятся числа n и k — количество точек с учениками, а также радиус зрения наблюдателя ($1 \leq n \leq 10^5, 1 \leq k \leq 2 \cdot 10^6$).

В следующих n строках вводятся по два числа — c_i и x_i — количество учеников, а также точка, в которой они находятся соответственно ($1 \leq c_i \leq 10^4, 0 \leq x_i \leq 10^6$). Гарантируется, что все x_i попарно различны.

Формат выходных данных

В единственной строке выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
4 5 4 11 20 100 10 7 7 3	21

Замечание

Рассмотрим первый тест. Есть 4 точки с учениками, а радиус зрения равен 5:

- 7 людей в точке 3
- 10 людей в точке 7
- 4 человека в точке 11
- 20 людей в точке 100

Наблюдатель может встать в точке 7. Тогда он будет видеть людей в точках 3, 7, 11, суммарно 21 ученик. Также он может встать в точке 8 или 6 и будет видеть столько же детей.

[C-B'] 3. Украсть пароль

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

Боб пытается взломать пароль Алисы от ejudge. Он знает что паролем всегда является строка-палиндром состоящая из строчных латинских букв.

Однажды Боб увидел как она набирает пароль и он записал его. Но он мог ошибиться. Строка очень похожа на пароль, и Боб решил, что случайно перепутал местами не более двух символов.

С этой задачей он и обратился к вам. Помогите — скажите, можно ли из записанной строки получить палиндром таким способом.

Формат входных данных

В единственной строке входного файла содержится строка s ($1 \leq |s| \leq 10^5$), состоящая из строчных латинских букв.

Формат выходных данных

В единственной строке выходного файла выведите «YES», если из строки s можно получить палиндром, поменяв не более двух символов местами и «NO» в противном случае.

Примеры

стандартный ввод	стандартный вывод
abacaba	YES
ab	NO
abbac	NO

[C-B'] 4. А какие бывают даты?

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

В серии Смешариков «Храм в огороде», когда герои нашли храм, им стало интересно какого века эта постройка. Они заметили табличку, на которой была дата постройки в формате $aa/bb/cc$. Оказалось, что постройка XXI века.

После анализа таблички Лосяш пришел к выводу, что дата написана по «григорианскому» календарю — год продолжительностью 365 дней, разделялся по нему на двенадцать месяцев. Второй месяц в году имел продолжительность двадцать восемь дней, первый, третий, пятый, седьмой, восьмой, десятый и двенадцатый — тридцать один день, остальные — тридцать дней. В особые года, номер которых делился на четыре и не делился на сто, либо делился на четыреста, второй месяц длился двадцать девять дней. Веком номер i назывался период с $100 \cdot (i - 1) + 1$ года по $100 \cdot i$.

Герои поняли, что не знают в каком порядке архитектор храма записывал даты. Например в России принято записывать в порядке день, месяц, год, а в США — месяц, день, год.

Лосяш бы и сам справился, но он попросил вас определить, каким датам в XXI веке могла соответствовать надпись, в предположении, что одно из чисел соответствует дню в месяце (дни в каждом месяце нумеровались с единицы), еще одно из чисел — номеру месяца (месяцы также нумеровались с единицы), и еще одно число — последним двум цифрам года в XXI веке григорианского календаря.

Формат входных данных

Во входном файле в формате $aa/bb/cc$ записаны числа с таблички.

Формат выходных данных

Выведите в произвольном порядке выведите все корректные даты $dd/mm/yy$, где dd соответствует номеру дня, mm — номеру месяца, yy — номеру года, причем числа, соответствующие dd , mm и yy являются перестановками чисел с таблички.

В случае, если никакая перестановка исходных чисел не является корректной датой в XXI веке, выведите «No such date».

Примеры

стандартный ввод	стандартный вывод
55/55/55	No such date
29/02/00	No such date
29/02/04	02/04/29 04/02/29 29/02/04 29/04/02

Замечание

Во втором примере по части «00» можно понять, что это может быть только годом. В XXI веке такой год — 2100. Этот год не високосный, значит в нем не может быть даты 29 февраля.

[C-B'] 5. Акулы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

На распродаже в ПЛЕА появились знаменитые акулы. Алиса хочет делать массовые закупки в течение n дней, пока идет распродажа. Алиса знает, что акул ровно X и они имеют товарные номера от 1-го до X -го.

В каждый из дней есть ограничение на число акул, которые можно приобрести – в день i нельзя купить менее a_i акул и нельзя купить более b_i . При этом можно покупать только отрезок акул с подряд идущими товарными номерами (например, можно купить акулы с 3-й по 6-ю, а вот отдельно 3-ю и 6-ю в один день нельзя). Также для упорядоченности организаторы распродажи хотят, чтобы номера проданных акул в день i были строго меньше номеров проданных акул в день $i + 1$. То есть акулы будут продавать строго слева направо.

Суммарно Алиса хочет купить все X акул. Помогите ей и скажите, каких акул в каждый из дней надо купить.

Формат входных данных

В первой строке дано число дней n и количество акул X ($1 \leq n \leq 10^5, 1 \leq X \leq 10^9$). В следующих n строках описаны ограничения по каждому дню – два числа a_i и b_i ($0 \leq a_i, b_i \leq 10^9$)

Формат выходных данных

Если купить ровно X акул невозможно, то выведите -1 .

Если же покупка возможна, то выведите n строк, где в i -й строке описан диапазон акул, которых надо купить в день i . Диапазон описывается двумя числами l_i и r_i – начало и конец покупаемого отрезка, границы включительные, нумерация с единицы.

Если в какой-либо день надо купить ноль акул, при этом последняя купленная акула имела номер x , то диапазон для дня с нулем акул будет иметь вид $[x + 1, x]$. Допустим, в первый день надо было купить 3 акулы, во второй и третий ноль, а в четвертый две. Тогда ответ будет иметь вид:

```
1 3
4 3
4 3
4 5
```

Если возможных ответов несколько и первый день, в который они различаются – это день i , то надо вывести тот ответ, в котором в день i куплен больший отрезок акул.

Примеры

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

Замечание

В первом примере Алиса может взять только одну акулу в первый день, во второй день она возьмет три акулы, а в третий шесть. Есть второй вариант – мы могли бы взять 1 акулу в первый день, 2 во второй и 7 в третий, но так как в первом ответе во второй день куплено три акулы, а не две, то мы берем первый вариант в качестве ответа.

Во втором примере Алиса не может купить менее шести акул в каждый из дней, поэтому 10 акул ей не набрать.

[C-B'] 6. Каждому — по набору задач

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

Саша, Коля и Тимур участвуют в командной олимпиаде по программированию. Составители задач олимпиады поработали на славу, и все задачи отранжированы по сложности, причём рост сложности между соседними задачами одинаков для всех задач. Без ограничения общности можно считать, что если задач n , то их сложности — это целые числа от 1 до n , где каждое встречается ровно один раз.

Ребята хотят занять первое место, поэтому им важно правильно распределить задачи между собой. Каждый из ребят способен решить каждую из задач за время, равное её сложности. Поскольку парни не новички в спортивном программировании, они быстро догадались, что оптимальным решением будет разделить задачи так, чтобы суммарная сложность задач, доставшихся каждому из них, была одинаковой. Если же это невозможно, то ребятам надо придумать другое распределение, но это уже совсем другая история.

Вам предстоит помочь ребятам распределить между собой задачи!

Формат входных данных

На вход вам подаётся единственное число $1 \leq n \leq 10^5$ — количество задач в констесте.

Формат выходных данных

Выведите для каждого из ребят набор задач, которые ему нужно решать. Наборы выводите следующим образом: В первой строке выведите количество задач в наборе. Далее, во второй строке, выведите сложности задач через пробел.

Если разбить задачи на три равных по суммарной сложности множества нельзя, выведите -1.

Если существует несколько решений, можно вывести любое.

Обратите внимание на то, что каждое число от 1 до n должно встречаться среди задач ровно один раз.

Примеры

стандартный ввод	стандартный вывод
5	2 1 4 2 2 3 1 5
7	-1
8	2 8 4 2 7 5 4 1 2 3 6

[В'-В] 1. Списать у списавшего — тоже преступление

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

На дворе 200022 год. Кружок Tinkoff Generation существует уже больше тысячи лет. Уже несколько сотен поколений школьников прошли через кружок, решая одни и те же задачи. И если в начале кружка преподаватели постоянно ловили школьников на списывании друг у друга, то со временем это сошло на нет. Но в какой-то момент преподаватели догадались, что вместо того, чтобы списывать друг у друга, школьники научились находить решения прошлых лет и списывать их. Разумеется, любой вид списывания на кружке запрещён, как и 100 лет назад, поэтому преподаватели решили разобраться с этой проблемой. Наказать школьников, которые списывали тысячу лет назад уже не получится, поэтому преподаватели решили проанализировать историю списываний, чтобы лучше понять паттерны и улучшить механизм проверки на списывание.

Для анализа истории списываний, преподаватели выбрали одну задачку, взяли все успешные решения по этой задаче за всю историю и выявили те из них, которые наиболее друг на друга похожи. Кроме того, благодаря грамотной психологической работе, преподаватели выяснили, что школьники все эти годы координировались между собой, вследствие чего никакое решение не было списано более двух раз. Более того, чтобы уменьшить вероятность быть обнаруженными, школьники списывали решение, которое было отправлено не менее чем k лет назад.

Теперь преподавателям стало интересно: имея n решений задачи, про каждое из которых известно, в каком году оно было отправлено, могло ли так оказаться, что все эти n решений были списаны с одного единственного решения по правилам, установленным заговорщиками. Увы, преподавателям надо готовиться к лекции, а это значит, что решить эту задачу предстоит вам.

Формат входных данных

Первая строка входного файла содержит два числа n и k — количество решений, имеющихся у преподавателей, и минимальное количество лет, которое должно пройти перед тем, как решение можно безопасно списать ($1 \leq n \leq 10^5$, $1 \leq k \leq 10^8$). Следующая строка содержит n чисел: a_1, a_2, \dots, a_n , где a_i ($0 \leq a_i \leq 10^9$) — сколько лет назад было отправлено i -е решение.

Формат выходных данных

Для каждого решения выведите номер решения, с которого оно было списано. Помните, каждое решение могли списать не более двух раз.

Если решение является оригинальным, то выведите для него число 0.

Если возможных решений несколько, выведите любое.

Если механизм проверки на списывание ошибается и такая история списываний невозможна, выведите единственное число -1 .

Примеры

стандартный ввод	стандартный вывод
6 3 2 10 6 0 5 2	3 0 2 5 2 3
4 3 10 1 1 1	-1

[В'-В] 2. Рома и примеры

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

Недавно Рома перешел в седьмой класс одной замечательной школы! И там, о ужас, ему пришлось раскрывать скобки. К сожалению, Роман пропускал занятия в шестом классе, поэтому не научился этого делать...

Однако он не отчаивается и уже побежал изучать этот материал. Примеры в седьмом классе сложные, поэтому часто бывает так, что уровень вложенности скобок даже больше трех! Поэтому он решил следующим образом изменить данное ему выражение: все скобки первого уровня он заменит на фигурные, все скобки второго — на квадратные, а остальные оставит круглыми. Однако сейчас он читает учебник, поэтому эту задачу придется выполнять вам!

Скобки первого уровня — это такие пары скобки, которые не вложены ни в какие другие. Скобки второго уровня — это те пары, которые ровно один раз вложены в скобки первого уровня, а также не вложены ни в какие другие. Все остальные скобки считаются третьего, четвертого и так далее уровня. Например, из выражения $((()))(())$ получится $\{[(())]\}[[]]$.

Формат входных данных

В первой и единственной строке вводится выражение, данное Роману. Гарантируется, что там будут только маленькие латинские буквы, круглые скобки, пробелы, цифры, а также следующие знаки: +, -, /, *. Также гарантируется, что круглые скобки образуют правильную скобочную последовательность. Длина выражения не превосходит 10^5 .

Формат выходных данных

Выведите, какая строка получится после изменения.

Примеры

стандартный ввод	стандартный вывод
$1 * (2 + (10 * (a + z)))$	$1 * \{2 + [10 * (a + z)]\}$
$(a + (a + (a + (a + (a + 2))))))$	$\{a + [a + (a + (a + (a + 2)))]\}$

[В'-В] 3. Игра

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

Вася и Петя играют в игру.

Изначально есть массив, отсортированный по возрастанию. Игроки делают ходы по очереди. За один свой ход игрок может вычесть из всех чисел на суффиксе массива единицу. Напомним, что суффиксом массива называется некоторое количество подряд идущих элементов, заканчивающееся последним элементом массива. Тот игрок, после чьего хода в массиве появляется два одинаковых числа или ноль, объявляется проигравшим.

Вася ходит первым, Петя — вторым. Кто одержит победу при оптимальной игре с обеих сторон?

Формат входных данных

В первой строке находится целое число n ($1 \leq n \leq 10^5$) — количество чисел в массиве.

Во второй строке через пробел записаны n различных целых чисел a_i ($1 \leq a_i \leq 10^9, a_{i-1} < a_i$).

Формат выходных данных

В единственной строке выведите **First**, если выиграет Вася и **Second**, если выиграет Петя.

Примеры

стандартный ввод	стандартный вывод
5 3 4 5 6 7	Second
1 1000	First

Замечание

В первом примере после первого хода массив может выглядеть следующим образом:

- (3, 4, 5, 6, 6)
- (3, 4, 5, 5, 6)
- (3, 4, 4, 5, 6)
- (3, 3, 4, 5, 6)
- (2, 3, 4, 5, 6)

Если Вася сделает один из первых четырех возможных ходов, игра сразу же закончится (победит Петя), иначе игра продолжится.

[B'-B] 4. GCD-MEX

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

Рассмотрим некоторое множество целых положительных чисел $\{x_1, x_2, \dots, x_k\}$. Определим функцию $f(\{x_1, x_2, \dots, x_k\})$ следующим образом: рассмотрим все непустые подмножества множества $\{x_1, x_2, \dots, x_k\}$ (всего существует $2^k - 1$ таких подмножеств), для каждого из них вычислим mex , а затем вычислим gcd всех полученных ранее mex -ов. Полученное число и будет значением функции $f(\{x_1, x_2, \dots, x_k\})$.

Рассмотрим вычисления данной функции на примере. Вычислим значение $f(\{1, 2, 4\})$. Для начала рассмотрим все непустые подмножества множества $\{1, 2, 4\}$, и для каждого из подмножеств вычислим его mex :

1. $\text{mex}(\{1\}) = 2$
2. $\text{mex}(\{2\}) = 1$
3. $\text{mex}(\{4\}) = 1$
4. $\text{mex}(\{1, 2\}) = 3$
5. $\text{mex}(\{1, 4\}) = 2$
6. $\text{mex}(\{2, 4\}) = 1$
7. $\text{mex}(\{1, 2, 4\}) = 3$

Теперь вычислим gcd всех полученных ранее mex -ов: $\text{gcd}(\{2, 1, 1, 3, 2, 1, 3\}) = 1$. Таким образом, $f(\{1, 2, 4\}) = 1$.

Перейдем непосредственно к условию задачи. Вам дан массив из n целых неотрицательных чисел a_1, a_2, \dots, a_n , а также q запросов. Каждый запрос задается парой чисел l и r ($1 \leq l \leq r \leq n$). Для каждого запроса вы должны вычислить значение $f(\{a_l, a_{l+1}, \dots, a_r\})$.

Напомним, что mex множества целых положительных чисел равен минимальному целому положительному числу, **не** входящему в множество.

Также напомним, что gcd множества целых положительных чисел равен наибольшему целому числу, на которое делятся все числа множества.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — количество элементов массива.

Вторая строка содержит n целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — элементы массива.

Третья строка содержит одно целое число q ($1 \leq q \leq 10^4$) — количество запросов.

Каждая из следующих q строк содержит два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — описание i -го запроса.

Формат выходных данных

Выведите q строк, в i -й строке выведите одно целое число — значение функции $f(\{a_{l_i}, a_{l_i+1}, \dots, a_{r_i}\})$.

Пример

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

Замечание

Рассмотрим пример из условия.

Для ответа на первый запрос нужно вычислить значение $f(\{a_1, a_2, a_3\}) = f(\{1, 2, 4\})$. Процесс вычисления функции с данными параметрами уже был описан выше в условии задачи.

Для ответа на второй запрос нужно вычислить значение $f(\{a_1\}) = f(\{1\})$. У множества $\{1\}$ существует ровно одно непустое подмножество, равное $\{1\}$. $\text{mex}(\{1\}) = 2$, $\text{gcd}(\{2\}) = 2$.

[В'-В] 5. Хорошая раскраска

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

Крош и Ежик красят скамейку. У них есть 5 цветов, а не только розовый.

Им стало скучно и они стали играть на ней в крестики-нолики. Расчертили всю скамейку на табличку с n строками и m столбцами. И стали играть.

После того как Крош проиграл, он расстроился и ушел. Ежик решил докрасить скамейку самостоятельно. Но ему хотелось сделать скамейку красивой. Красивой скамейка считается, если в любом квадрате 3×3 в «кресте» присутствуют все пять цветов, а в «рамке» не более четырех различных цветов, чтобы не сильно бросалась в глаза (под «крестом» будем понимать все клетки квадрата 3×3 , кроме угловых, а под «рамкой» все клетки квадрата, кроме центральной).

Ежик не силен в таких задачах, но ему очень нужно сделать это до прихода Кроша. Подскажите ему или скажите, что так сделать нельзя.

Формат входных данных

В первой и единственной строке входного файла дано два натуральных числа n, m ($1 \leq n, m \leq 100$).

Формат выходных данных

В выходной файл выведите nm чисел. j -е число в i -й строке ($1 \leq i \leq n; 1 \leq j \leq m$) означает цвет соответствующей клетки (цвет — натуральное число от 1 до 5 включительно).

Если существует несколько ответов, выведите любой. Если ответа не существует, выведите -1.

Пример

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

[В'-В] 6. Петергофские сады

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

Филипп приехал в Петергоф, чтобы прогуляться по местным паркам вдоль дворца и Финского залива. Его впечатлило наличие в парке прогулочного лабиринта. Таковые создавались в стародавние времена, чтобы вельможи, прогуливаясь в одном и том же месте, могли ходить разными маршрутами. Для Филиппа вполне естественно, что лабиринт можно задать как граф, где отдельные переходы представляют собой рёбра, а интересные точки лабиринта — вершины (причём переходы двусторонние, то есть перемещаться от одной интересной точки к другой по переходу можно в обоих направлениях). Вельможи прошлого много думали о своих нервах, а точнее думали об этом придворные проектировщики. А потому, чтобы вельможи не петляли кругами, в лабиринте нет циклов. Вельможи жили в большом дворце и (как понятно из назначения лабиринта) ходили до него одним маршрутом. Поэтому чтобы никакая часть лабиринта не была бесполезной, от любой точки лабиринта можно добраться до любой другой. Но, конечно, Петергофский нижний сад состоит не из одного лишь лабиринта. Так, например, в западной части ансамбля располагается чудеснейший розарий!

Реновация! Парк Петергофа будет обновлён под стать современным тенденциям. Узнав об этом, Филипп устроился одним из разработчиков плана реновации. Приехав на место, он долго думал, что можно поменять. Естественно, пока он думал, он бродил по лабиринту. Естественно, лучшей идеей для перестроения стал лабиринт. Филипп решил соединить лучшее из того, что видел. Он решил добавить по одному розовому кусту из розария в каждую точку лабиринта.

Всего, по подсчётам Филиппа, в розарии содержится s сортов роз (розарий невероятно велик и, чтобы регулярно заменять засохшие кусты, в Петергофе хранятся бесконечные запасы кустов каждого из сортов). Чтобы посетителям не было скучно, Филипп решил, что по плану рассадки розовые кусты в точках, кратчайший путь между которыми состоит из не более чем двух переходов лабиринта, должны быть разных сортов.

Филипп умён, уже окончил школу дважды и собирается сделать это в третий раз. Потому он понял, что способов рассадить розы в лабиринте может быть много. Он посчитал это число в голове, но боится, что мог ошибиться, поэтому просит Вас написать программу, которая посчитает это число по модулю $10^9 + 7$ (он знает, что способов может быть реально много и ему нужно просто удостовериться в своей правоте).

Формат входных данных

В первой строке входных данных содержатся два числа: n и s ($1 \leq n, s \leq 10^5$) — количество интересных точек в лабиринте и сортов роз в розарии, соответственно.

В следующих $n - 1$ строках содержатся по два числа u_i и v_i ($1 \leq u_i, v_i \leq n$), означающие, что между интересными точками номер u_i и v_i есть переход. Гарантируется, что они задают корректный по условию задачи лабиринт.

Формат выходных данных

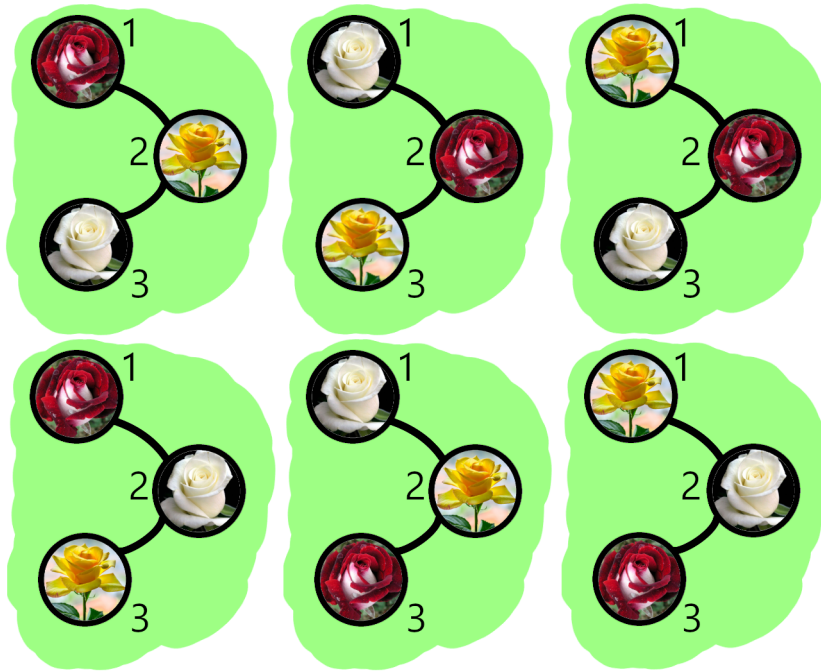
Выведите одно число — количество различных способов рассадить в интересных точках лабиринта по розовому кусту так, чтобы удовлетворить требованиям плана Филиппа, взятое по модулю $10^9 + 7$. Два способа считаются различными если хотя бы в одной точке лабиринта в этих двух вариантах планируется посадить кусты разных сортов.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 3	6
2 5 1 2	20

Замечание

Ниже представлены все варианты рассадки роз для первого примера.



6 вариантов

[В-А'] 1. Клуб любителей строк

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

Второклассник Иван всегда мечтал попасть в клуб любителей строк. Чтобы успешно пройти отбор, необходимо справиться со вступительным заданием. Иван любит строки, а не задачи, поэтому обратился к Вам за помощью.

Дана строка s , состоящая из строчных латинских символов. Вхождением строки t назовем пару чисел (l, r) ($l \leq r$), что подстрока строки s с l по r символ включительно совпадает со строкой t . Скажем, что вхождения (l_1, r_1) и (l_2, r_2) - непересекающиеся, если отрезки $[l_1; r_1]$ и $[l_2; r_2]$ не пересекаются. Красоту строки t определим как максимальное количество ее попарно непересекающихся вхождений в качестве подстроки в строку s . Среди всех строк с максимальной красотой необходимо найти самую длинную строку и сообщить главе клуба ее размер.

Формат входных данных

В первой и единственной строке дана строка s ($1 \leq |s| \leq 10^5$), состоящая только из строчных латинских букв.

Формат выходных данных

В единственной строке выходного файла выведите максимальную длину строки, красота которой максимальна.

Примеры

стандартный ввод	стандартный вывод
хухзхух	1
baba	2

Замечание

В первом примере максимальную красоту имеет строка x . Её длина - 1.

Во втором примере максимальную красоту имеют строки a , b и ba . Размер самой длинной из них - 2.

[B-A'] 2. Село

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

Сеть дорог в село Бендереевое представляет из себя граф из n вершин и некоторого количества ребер. В вершине с номером 1 находится главный корпус, в котором находится большое количество полезных ресурсов. Бармалей решил, что было бы неплохо перекрыть какую-то из дорог так, чтобы у как можно большего числа вершин их кратчайшее расстояние до вершины с номером 1 изменилось (включая те, которые станут недостижимы). Бармалей просит вас посчитать максимальное количество таких вершин.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 300$) — количество вершин графа. В следующих n строках содержатся по n целых чисел $a_{i,j}$ ($-1 \leq a_{i,j} \leq 10^9$) — если в графе не существует ребра, то $a_{i,j} = -1$, иначе $a_{i,j}$ — длина дороги, соединяющей вершины i и j . Гарантируется, что $a_{i,j} = a_{j,i}$, $a_{i,i} = 0$

Формат выходных данных

Выведите максимальное количество вершин, расстояние от которых до вершины с номером 1 увеличится, если мы перекроем 1 дорогу.

Пример

стандартный ввод	стандартный вывод
5 0 3 5 1 2 3 0 10 1 -1 5 10 0 1 -1 1 1 1 0 3 2 -1 -1 3 0	3

Замечание

В примере из условия мы можем перекрыть дорогу, соединяющую вершины с номерами 1 и 4. Так расстояние до вершин 4, 3 и 2 увеличится.

[B-A'] 3. Планирование учебного года

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

Уже скоро начнется новый учебный год, и Антон задумался о том, как он хочет его провести. Мальчик считает, что все в этой жизни нужно планировать, в том числе и свои оценки, поэтому он заранее составил для себя несколько правил, которым он будет безукоризненно следовать в течение всего учебного года.

Антон считает, что он должен получать только оценки 3, 4 и 5. Более того, ему известно, что за весь учебный год состоится n уроков математики, за каждый из которых он получит одну из трех возможных оценок. Антон считает, что год будет удачным, если будут выполнены m условий.

Каждое из m условий задается тремя целыми числами l_i , r_i и x_i ($1 \leq l_i \leq r_i \leq n$, $1 \leq x_i \leq 3$) и гласит, что среди оценок, полученных Антоном на l_i -м, $(l_i + 1)$ -м, ..., r_i -м уроках математики должно быть ровно x_i различных оценок.

Так как Антон любит математику, он тут же решил посчитать, сколько существует способов заработать оценки на каждом из n уроков таким образом, чтобы все m условий были выполнены. А теперь решить данную задачу придется и вам.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 300$) — количество уроков математики, которые состоятся в учебном году, и количество условий, которые должен выполнить Антон.

Каждая из следующих m строк содержит три целых числа l_i , r_i и x_i ($1 \leq l_i \leq r_i \leq n$, $1 \leq x_i \leq 3$) — описание i -го условия, которое должен выполнить Антон.

Формат выходных данных

Выведите одно целое число — количество способов заработать оценки на каждом из n уроков таким образом, чтобы все m условий были выполнены.

Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 2 1 3 3 1 2 1	0
3 3 2 2 1 1 3 2 2 3 1	6

Замечание

Рассмотрим первый пример из условия. Всего за год состоится три урока математики, причем Антон должен выполнить два условия:

1. Среди оценок, полученных за уроки 1, 2 и 3, должно быть ровно три различных оценки.
2. Среди оценок, полученных за уроки 1 и 2, должна быть ровно одна различная оценка.

Нетрудно понять, что эти два условия противоречат друг другу, поэтому количество способов заработать оценки равно нулю.

Во втором примере из условия подходят следующие способы получить оценки:

1. [1, 2, 2]
2. [1, 3, 3]
3. [2, 1, 1]
4. [2, 3, 3]
5. [3, 1, 1]
6. [3, 2, 2]

[В-А'] 4. Покраска графа

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

Вам дан неориентированный граф (иначе говоря, по каждому ребру можно проходить в двух направлениях), состоящий из N вершин и M ребер.

Изначально все вершины покрашены в цвет 0. После этого вы начнете красить вершины графа.

На i -м шаге вы покрасите в цвет c_i все вершины, которые находятся на расстоянии не более чем d_i от вершины u_i . Расстоянием между вершинами называется минимальное число ребер на пути между ними.

Теперь вы хотите узнать цвет всех вершин после такой покраски.

Формат входных данных

В первой строке вводится два целых числа N, M ($1 \leq N, M \leq 10^5$) — число вершин и ребер в графе, соответственно.

В следующих M строках вводится по два целых числа a_i, b_i ($1 \leq a_i, b_i \leq N$) — описание ребер графа — i -е ребро соединяет вершины a_i и b_i .

В следующей строке вводится одно целое число Q ($1 \leq Q \leq 10^5$) — число шагов покраски.

В следующих Q строках вводится по три целых числа u_i, d_i, c_i ($1 \leq u_i \leq n, 0 \leq d_i \leq 10, 1 \leq c_i \leq 10^5$) — описание шагов покраски.

Обратите внимание на ограничения на d_i

Формат выходных данных

Выведите N чисел — цвета вершин $1, 2, \dots, N$ после такой покраски.

Пример

стандартный ввод	стандартный вывод
3 3	5
1 2	3
2 3	5
1 3	
2	
1 3 5	
2 0 3	

Замечание

В первом примере цвета вершин меняются так:

1. До всех шагов цвет всех вершин равен 0
2. После первого шага цвет всех вершин равен 5
3. После второго шага цвет вершин 1 и 3 равен 5, а вершины цвет вершины 2 равен 3

[В-А'] 5. Па-па-палиндромы

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

Вам дана строка s , состоящая из латинских символов.

Определим $next(c)$ как функцию, принимающую символ и возвращающую следующий символ в алфавите (если следующего символа нет, то возвращается первый символ алфавита). Например, $next(\langle a \rangle) = \langle b \rangle$, $next(\langle g \rangle) = \langle h \rangle$, $next(\langle z \rangle) = \langle a \rangle$.

Определим $f(s)$ как количество таких подмножеств индексов строки s , из символов на которых с помощью перестановки возможно собрать строку-палиндром (иначе говоря, строку, равную самой себе, записанной в обратном порядке). Например, $f(\langle aab \rangle) = |\{[0], [1], [2], [0, 1], [0, 1, 2]\}| = 5$, а соответствующие строки-палиндромы равны $\langle a \rangle$, $\langle a \rangle$, $\langle b \rangle$, $\langle aa \rangle$, $\langle aba \rangle$.

От вас требуется ответить на q запросов, каждый одного из двух типов:

- 1 $i j k$ — применить ко всем символам s_i, s_{i+1}, \dots, s_j операцию $next$ k раз подряд.
- 2 $i j$ — вычислить значение $f(s_i s_{i+1} \dots s_j)$ (то есть f от подстроки $s_{i..j}$). Так как получившееся число может быть очень большим, от вас требуется посчитать его по модулю $10^9 + 7$ (иначе говоря, остаток от $f(s_i s_{i+1} \dots s_j)$ при делении на $10^9 + 7$).

Формат входных данных

Первая строка содержит два натуральных числа n, q ($1 \leq n, q \leq 10^5$) — длину s и количество запросов соответственно.

Вторая строка содержит строку S из строчных латинских символов. Гарантируется, что $|s| = n$.
Следующие q строк содержат запросы. Формат запросов описан в условии.

Формат выходных данных

Для каждого запроса второго типа выведите одно целое число - ответ на данный запрос. Не забудьте, что проверяется не значение $f(s_i \dots s_j)$, а его остаток при делении на $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
3 3	5
aab	2
2 0 2	
1 0 0 2	
2 0 1	

Замечание

В первом запросе требуется вычислить $f(\langle aab \rangle)$, этот запрос был прокомментирован в тексте условия.

Второй запрос изменит строку на $\langle sab \rangle$.

В третьем запросе требуется вычислить $f(\langle sa \rangle) = |\{[0], [1]\}| = 2$. В этой строке нет подмножеств, которые бы формировали палиндром, кроме каждой буквы по отдельности.

[В-А'] 6. Палиндромный ребрендинг

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

Независимая общественная интернет-организация «Фанаты дерева палиндромов» старается расширять использование палиндромов в повседневной жизни. Чтобы убедить компании сделать свое название палиндромом, они решили создать программу для автоматического ребрендинга.

Пусть предыдущее название компании это строка s , состоящая из строчных символов латинского алфавита. Тогда программа удалит несколько подряд идущих символов строки s (в любом месте, возможно ноль). Оставшаяся строка будет новым названием компании. Естественно, она должна быть палиндромом. Среди всех возможных новых названий будет выбираться наибольшее по длине (если таких несколько, то любое).

Реализуйте описанную программу для ребрендинга и помогите любимым палиндромам захватить мир!

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Следующие t строк описывают наборы входных данных.

Для каждого набора входных данных, единственная строка содержит строку s ($1 \leq |s| \leq 10^6$), состоящую из строчных символов латинского алфавита.

Гарантируется, что сумма длин строк s по всем наборам входных данных не превосходит 10^6 .

Формат выходных данных

Для каждого набора входных данных выведите новое возможное название компании после ребрендинга. Если существует несколько возможных вариантов, вы можете вывести любой.

Пример

стандартный ввод	стандартный вывод
7	ffffff
ffffff	ff
tinkoff	a
algocode	aboba
aboba	anana
anas	sus
suslolmemkek	roflrrlfor
roflrroofflllfor	

Замечание

В первом наборе входных данных, название уже является палиндромом, поэтому не нужно удалять символы.

Во втором наборе входных данных можно удалить символы с 1-го по 5-й.

В седьмом наборе входных данных можно удалить символы с 7-го по 12-й.

[A'-A] 1. Интересные операции

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

После длительной прогулки по лесу ежик нашел строку s , состоящую из маленьких букв латинского алфавита. Тут ежик вспомнил, что раньше он коллекционировал палиндромы. Ему стало интересно, можно ли сделать из строки s палиндром последовательными обменами двух соседних символов. Если же это возможно, то ежик хочет знать минимальное количество обменов, необходимое для достижения данной цели.

Формат входных данных

В первой и единственной строке входных данных содержится строка s ($1 \leq |s| \leq 2 \cdot 10^5$), состоящая из маленьких букв латинского алфавита.

Формат выходных данных

Если ежик не может получить палиндром из строки s , выведите -1 , иначе выведите минимальное необходимое количество обменов соседних символов, чтобы из s получить палиндром.

Примеры

стандартный ввод	стандартный вывод
aboba	0
generation	-1
boaba	2
uuxwszsuxuwz	9

Замечание

В первом тестовом случае строка s уже является палиндромом, поэтому нам не нужно выполнять никаких операций.

Во втором случае можно показать, что какие бы обмены мы не выполняли, строка никогда не станет палиндромом.

В третьем случае мы можем сначала сделать обмен **bo**аbа и получить строку баоbа, затем сделать обмен **ba**оbа и получить палиндром.

[A'-A] 2. Весело и точка

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

Костя на свой день рождения устраивает пикник на поляне, которая представляет из себя клетчатый прямоугольник $n \times m$.

Для того, чтобы приукрасить поляну, Костя заказал услуги компании «Весело и Точка», которая обещала развесить на поляне Висюльки и разложить Тарелки.

Согласно плану Кости, в каждой клетке может быть либо Висюлька, либо Тарелка (может не быть ничего).

Менеджер компании «Весело и точка» считает, что украшать поляну нужно постепенно. А именно, грузчики будут проходить из левого верхнего угла поляны в правый нижний, раскладывая Тарелки или развешивая Висюльки. Один и тот же грузчик не может одновременно и развешивать Висюльки и раскладывать Тарелки.

Грузчик может пройти от левого верхнего угла до правого нижнего, не обязательно кратчайшим путем (возможно, посещая одну и ту же клетку несколько раз), переходя в соседние по сторонам клетки. В каждой посещенной клетке грузчик повесит Висюльку или поставит Тарелку (зависит от того, с чем этот грузчик шел). Если в посещенной клетке уже была Висюлька или Тарелка, то произойдет замена — старый предмет выкинут.

Менеджеру компании интересно, какое минимальное количество человек понадобится, чтобы реализовать план Кости. Помогите ему с вычислениями.

Формат входных данных

В первой строке вводятся два числа n, m ($1 \leq n, m \leq 1000$) — размеры поляны.

В следующих n строках вводится по m символов — описание плана Кости. В клетках с символом «В» (ASCII-код 66) должна находиться Висюлька, в клетках с символом «Т» (ASCII-код 84) — тарелка. В клетках с символом «*» (ASCII-код 42) не должно находиться ничего.

Обратите внимание, что символы «В» и «Т» являются латинскими заглавными буквами.

Формат выходных данных

Выведите минимальное количество грузчиков, необходимое для реализации плана. Гарантируется, что план реализуем.

Пример

стандартный ввод	стандартный вывод
5 3 ТТ* *Т* ВТВ *ТТ **Т	2

Замечание

В тесте из условия сначала по поляне прошел грузчик, и повесил 9 Висюлек, а затем прошел грузчик с Тарелками, который выкинул все Висюльки, кроме 2, и поставил вместо них Тарелки.

[A'-A] 3. Оупен-спейс

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

Очередная it-компания арендует помещение, являющееся клетчатым прямоугольником $n \times m$. Каждая сторона между клетками является стеной, образуя таким образом $n \cdot m$ комнат, в которых программисты горбятся над своими тусклыми мониторами.

Новый менеджер решил поменять планировку этажа и снести k стен между комнатами, чтобы сделать пространство «оупен-спейсом».

У каждой комнаты есть координаты (x, y) , соответствующие декартовой системе координат над клетчатым прямоугольником. Порядок нумерации можно сверить со примером схемы этажа.

Назовем «закрытым пространством» такой подпрямоугольник, который со всех сторон окружен стенами, а также не содержит никаких стен внутри. Например, комната в исходном прямоугольнике является закрытым пространством, так же, как и две комнаты в исходном прямоугольнике, между которыми снесли стену.

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

В примере схемы есть 9 закрытых пространств — 8 комнат 1×1 и одно пространство 2×2 , внутри которого нет стен.

От вас требуется после сноса каждой стенки находить количество «закрытых пространств» на текущий момент.

Формат входных данных

Первая строка содержит два числа n, m ($1 \leq n, m \leq 10^5$) — размеры помещения.

Во второй строке вводится число k ($1 \leq k \leq 10^5$) — количество удаляемых стен.

В следующих k строках вводится по 4 числа x_a, y_a, x_b, y_b ($0 \leq x_a, x_b < n, 0 \leq y_a, y_b < m$) — координаты комнат, стена между которыми должна быть снесена. Гарантируется, что комнаты являются соседними.

Формат выходных данных

Выведите k натуральных чисел — количество «закрытых пространств» после удаления текущей стены.

Пример

стандартный ввод	стандартный вывод
3 3	8
5	6
0 0 0 1	5
0 0 1 0	6
1 0 1 1	5
0 1 1 1	
2 2 2 1	

Замечание

В первом тесте в результате всех операций есть 3 комнаты 1×1 ($[(0, 2), (2, 0), (2, 1)]$), 1 пространство 2×2 (левый верхний угол в $(0, 0)$) и 1 пространство 2×1 (левый верхний угол в $(1, 2)$).

[A'-A] 4. Мишени от Мишани

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

Ваш друг Миша позвал вас в тир стрелять по мишеням. В тире есть n мишеней, расположенных вдоль стены. За попадание в i -ю в порядке «слева направо» мишень начисляется a_i очков.

Миша — неуверенный в себе стрелок, который боится промахнуться. А именно, он не будет стрелять в мишень, если хотя бы одна из соседних мишеней слева или справа от нее была сбита. Несмотря на свою неуверенность, Миша очень хорошо стреляет, и всегда попадает в цель, сбивая мишень, в которую он целился.

На протяжении вашего посещения тира мишени меняли q раз. На i -й раз мишень с номером x_i получала новую ценность в d_i очков.

После каждой замены Миша спрашивает — сколько он может набрать очков, если собьет самое ценное множество мишеней? Помогите ему разобраться.

Формат входных данных

В первой строке вводятся два целых числа n, q ($1 \leq n, q \leq 50\,000$) — количество мишеней и число запросов соответственно.

В следующих n строках вводится по одному целому числу a_i ($1 \leq a_i \leq 100\,000$) — ценность мишени i .

В следующих q строках вводится по два целых числа x_i и d_i ($1 \leq x_i \leq n, 1 \leq d_i \leq 100\,000$) — описание запроса изменения мишени на позиции x_i , ее новая ценность равна d_i .

Формат выходных данных

Выведите q чисел — максимальное количество очков, которое Миша может набрать в тире, после каждого из запросов.

Пример

стандартный ввод	стандартный вывод
4 3	4
1	4
1	102
1	
1	
2 3	
3 2	
1 100	

Замечание

В первом примере примере мишени будут меняться таким образом:

- После первого запроса мишени будут стоять $[1, 3, 1, 1]$ очков. Миша может сбить 2 мишени и получить 4 очка: $([1, \underline{3}, 1, 1])$
- После второго запроса мишени будут стоять $[1, 3, 2, 1]$ очков. Стратегия Миши не меняется.
- После третьего запроса мишени будут стоять $[100, 3, 2, 1]$ очков. Теперь Миша может получить 102 очка $([\underline{100}, 3, 2, 1])$

[A'-A] 5. Максимальное покрытие

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

Вам дано N отрезков. Вы хотите удалить ровно K отрезков из этого множества так, чтобы размер объединения оставшихся был максимален. Размером объединения отрезков называется суммарная длина числовой прямой, которая покрыта хотя бы одним из отрезков. Например, размер объединения отрезков $\{[2, 4], [3, 5]\}$ равен 3.

Формат входных данных

В первой строке вводится два целых числа N и K ($1 \leq N \leq 10^5, 1 \leq K \leq \min(100, N)$) — количество отрезков, и число отрезков, которые нужно удалить.

В следующих N строках вводятся по два целых числа L_i и R_i ($1 \leq L_i \leq R_i \leq 10^9$) — начало и конец i -го отрезка.

Формат выходных данных

Выведите одно число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
5 1 1 11 5 20 100 103 200 210 205 220	39

Замечание

В первом примере если удалить третий отрезок, то объединение оставшихся состоять из двух отрезков $[1, 20]$ и $[200, 220]$, их суммарная длина равна 39. Можно показать, что удалять третий отрезок оптимально.

[A'-A] 6. Построение на физкультуре

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

На уроке физкультуры n учеников должны встать в ряд по возрастанию роста. Учителю показалось, что это слишком скучно, поэтому он придумал процесс, в ходе которого ученики должны постараться встать как можно более правильно.

Изначально они встали в ряд в порядке p_1, p_2, \dots, p_n , где p это некоторая перестановка целых чисел от 1 до n , p_i означает номер человека в классе по росту (считаем, что у всех учеников в классе разный рост, поэтому их можно однозначно упорядочить по росту).

Затем, $\frac{n}{2}$ раз (гарантируется, что n четное число) некоторая пара подряд стоящих учеников должна выйти вперед и встать в начало нового ряда. В старом ряду все ученики сдвигаются так, что ученики, которые стояли по краям от вышедшей вперед пары становятся соседними.

Учитель считает построение успешным, если перестановка, полученная в новом ряду является лексикографически минимальной среди всех перестановок, которые можно было получить в ходе какого-нибудь корректного процесса построения. Помогите ученикам найти новую лексикографически минимальную перестановку, в которую они могут успешно выстроиться.

Формат входных данных

В первой строке находится единственное целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество детей в классе. Гарантируется, что n четное.

Во второй строке находится n различных целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — изначальный порядок учеников в ряду.

Формат выходных данных

Выведите n различных целых чисел — лексикографически минимальную перестановку, в которую могут выстроиться ученики.

Примеры

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

Замечание

В первом тесте ученики могут выстроиться следующим образом:

- Старый ряд: $(2, 3, \underline{6}, 4, 5, 1) \rightarrow (2, 3, 5, 1) \rightarrow (\underline{2}, 1) \rightarrow ()$
- Новый ряд: $() \rightarrow (6, 4) \rightarrow (3, 5, 6, 4) \rightarrow (2, 1, 3, 5, 6, 4)$

Во втором тесте ученики могут выстроиться следующим образом:

- Старый ряд: $(1, \underline{7}, 5, 4, 2, 3, 6, 8) \rightarrow (1, 4, 2, 3, \underline{6}, 8) \rightarrow (1, \underline{4}, 2, 3) \rightarrow (\underline{1}, 3) \rightarrow ()$
- Новый ряд: $() \rightarrow (7, 5) \rightarrow (6, 8, 7, 5) \rightarrow (4, 2, 6, 8, 7, 5) \rightarrow (1, 3, 4, 2, 6, 8, 7, 5)$

[A] 1. Шай-Хулуд: Возрождение легенды

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

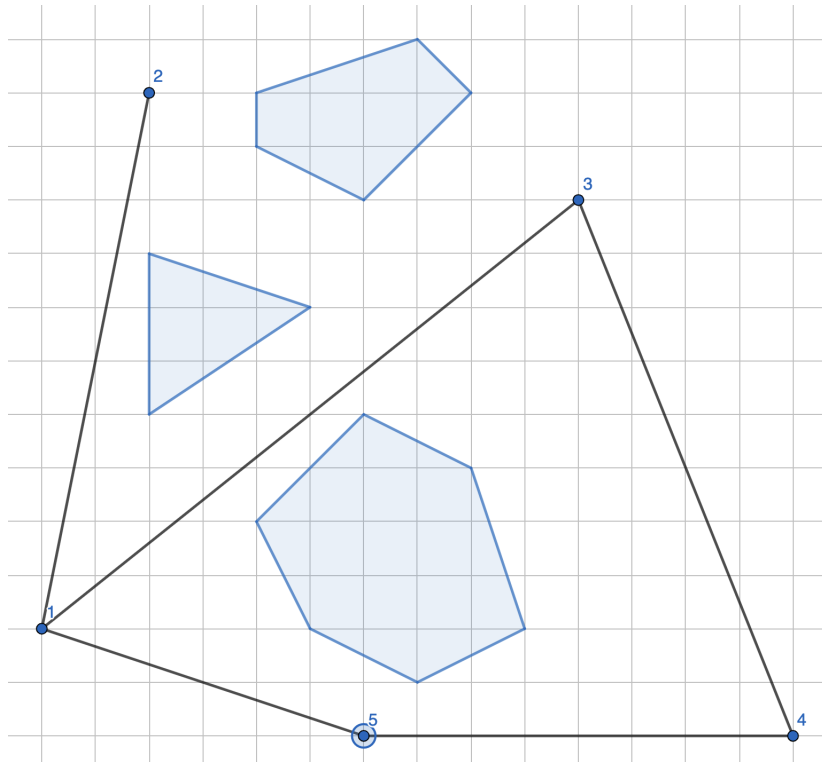
На планете Арракис обитают огромные песчаные черви, именуемые Шай-Хулуд. Они могут перемещаться по планете, по городам и дорогам, их соединяющим. Дорожная инфраструктура планеты очень специфична. Города связаны дорогами таким образом, что никакие две дороги не соединяют одну и ту же пару городов, никакая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

Недавно на планете Арракис решили сделать дорожную реформу. Было решено, что дорог слишком много, и поэтому часть из них надо перестать обслуживать. При этом чтобы сильно не портить привычную всем дорожную карту, было решено удалить не более одной дороги, выходящей из каждого города. Так же было важно оставить все города связными, то есть добиться того, чтобы из любого города можно было бы добраться до любого другого по дорогам страны.

Чтобы сделать такую дорожную реформу, было решено обратиться к вам, однако так как планета Арракис находится слишком далеко от вас, это сообщение до вас не дошло и дорожную реформу провели без вашей помощи.

Изменение дорожной карты очень сильно расстроило червей, проживающих на планете. Теперь им приходится двигаться другими маршрутами. Это настолько не понравилось Шай-Хулуд, что они начали игнорировать дороги и иногда двигаться не по дорогам.

Представим планету Арракис в виде плоскости. На планете есть k городов, i -й из них расположен в точке с координатами (x_i, y_i) . Между городами осталось t дорог, i -я из них соединяет города с номерами a_i и b_i . Гарантируется, что можно добраться от любого города до любого другого. Так же на планете есть l скал, каждая из них представляет собой выпуклый многоугольник. Шай-Хулуд могут двигаться по дорогам со скоростью v_i в секунду, а вне дорог со скоростью u_i метров в секунду. По внутренним точкам скал двигаться нельзя. На картинке ниже показан пример расположения городов, дорог и скал на планете:



Недавно два червя, находящиеся в городах с номерами q и w решили собраться в одном городе, чтобы поиграть там настольные игры. Для этого им потребовалось выбрать некоторый город, до

которого они доберутся. Черви хотят как можно раньше начать играть, а поэтому им требуется найти такой город, чтобы при движении по самому оптимальному пути, последний из них оказался в городе как можно быстрее. Для помощи с выбором города они обратились к вам, однако в этот момент планета Меркурий находилась в ретроградной фазе движения, а поэтому сообщение до вас не дошло.

Не дождавшись вашей помощи, Черви собрались в одном городе и начали играть настольные игры. Настольные игры червей весьма экзотичны, рассмотрим игру Червей в карты. В этой игре у каждого червя есть колода из 49 карт. На каждой карте написано число от 1 до 7, также каждая карта имеет одну из семи мастей (в скобках указана буква, которой масть обозначается во входных данных): Roman (R), Oleg (O), Yegor (Y), Gleb (G), Boris (B), Nikita (N) и Petr (P). Все карты уникальны, и в колоде присутствует комбинация каждой цифры с каждой мастью.

У каждого игрока есть карты в руке, карты, выложенные перед ним — его армия, и общая стопка карт — поле боя. Изначально у первого игрока есть n карт в руке и одна в армии, у второго — m карт в руке и одна в армии, а поле боя пустое, однако считается, что на нем лежит карта масти Roman (R).

На картах определен строгий порядок старшинства. Первая карта считается старше второй, если на ней написано большее число, или числа равны, но ее масть идет раньше в последовательности мастей, обозначенной выше, то есть $R > O > Y > G > B > N > P$. Таким образом, самая старшая карта в игре — семерка масти Roman, а самая младшая — единица масти Petr. Гарантируется, что в начале игры карта в армии первого игрока младше, чем карта в армии второго игрока.

Масть карты, лежащей верхней на поле боя, задает условие, по которому определяется лидирующий игрок. Эти условия таковы: лидирует игрок, у которого

- Roman (R) — самая старшая карта;
- Oleg (O) — больше карт с одинаковым значением;
- Yegor (Y) — больше карт одной масти;
- Gleb (G) — больше карт с четным значением;
- Boris (B) — больше карт различных мастей;
- Nikita (N) — больше карт с подряд идущими значениями (например, 4 – 5 – 6);
- Petr (P) — больше карт со значением меньше 4.

Чтобы определить лидирующего игрока, необходимо выбрать оптимальную комбинацию карт из армии каждого игрока, и сравнить их по выбранному условию. Если по данному условию комбинации совпадают (например, на поле боя лежит карта масти Gleb (G), и у обоих игроков по две карты с четным значением), тогда лидирует игрок, у которого самая старшая карта **среди карт комбинации** превосходит такую карту противника. Смотрите примеры и пояснения к ним для лучшего понимания этой механики.

Игроки делают ходы по очереди. В свой ход игрок должен совершить одно из трех действий:

- выложить одну карту из руки в свою армию,
- выложить одну карту из руки на поле боя,
- выложить одну карту из руки в свою армию, и выложить еще одну карту из руки на поле боя,

таким образом, чтобы после этого действия игрок, совершающий ход, лидировал (вероятно, по новому условию). Если игрок не может совершить ход таким образом, чтобы лидировать в конце хода, то игра заканчивается, и его противник выигрывает. В частности, игрок с пустой рукой проигрывает в начале своего хода, однако оставшись с пустой рукой в конце хода игрок еще может выиграть.

Обратим внимание, что по правилам игрок не может класть на поле боя карту, если у него в армии нет ни одной карты, удовлетворяющей условию карты, положенной на поле боя. Например,

если у игрока нет ни одной четной карты в армии, он не может положить карту масти Gleb (G) на поле боя.

Черви очень хотят узнать, кто же из них победит при правильной игре. Помогите им это сделать!

Формат входных данных

В первой строке заданы числа n и m — количество карт в руке у первого и второго игрока ($0 \leq n, m \leq 6$).

Вторая строка содержит описание $n + 1$ карты первого игрока, первая из которых изначально находится в армии, а остальные — в руке. Описание карты состоит из двух символов: цифры на карте d_i и ее масти c_i ($1 \leq d_i \leq 7, c_i \in \{R, O, Y, G, B, N, P\}$).

Третья строка содержит описание $m + 1$ карты второго игрока в таком же формате.

Формат выходных данных

Выведите одно слово: «First» (без кавычек), если выиграет первый игрок, и «Second» (без кавычек) — если выиграет второй.

Примеры

стандартный ввод	стандартный вывод
0 0 5R 6P	Second
3 0 1R 5O 7R 4G 4R	First
2 0 1R 2R 3R 7R	Second
4 5 1Y 3Y 1P 1N 2B 4G 3O 6Y 5O 1R 2R	Second

Замечание

Комбинацией карт считается множество, удовлетворяющее текущему условию. Чтобы выбрать оптимальную комбинацию, нужно сначала максимизировать ее размер, а затем — старшую карту. Например, у игрока в армии лежат карты 1G, 1R, 3O, 5P, 5O. Тогда его оптимальные комбинации для каждого условия таковы:

- Roman (R) — 5O;
- Oleg (O) — 5P + 5O, это лучше, чем 1G + 1R;
- Yegor (Y) — 3O + 5O;
- Gleb (G) — нет комбинации, а значит игрок не может претендовать на лидерство по этому условию;
- Boris (B) — 1G + 1R + 5P + 5O;
- Nikita (N) — 5O, комбинация для N условия может состоять из 1 карты;
- Petr (P) — 1G + 1R + 3O.

В первом примере ни у одного из игроков нет карт в руке, однако выигрывает второй игрок, так как первый игрок не может сделать ход.

Во втором примере рассмотрим какой масти окажется карта на поле после первого хода:

- Если масть R, то первому игроку надо иметь в армии карту более старшую карту. Тогда он может выложить карту 7R в армию, тогда у него будет два варианта комбинаций: 1R или 7R, из них более оптимальной будет 7R. Поле останется масти R и ход будет успешным для первого. Если он выложит карту 7R на поле, то в его единственной возможной комбинацией будет 1R, это младше карты второго игрока.
- Если масть O, то первому игроку надо будет выложить карту 5O на поле боя. Однако если только выкладывать карту 5O на поле, то в армии обоих игроков будут комбинации из одной карты с одинаковым значением, однако у второго игрока карта старше. Если же выложить карту 5O на поле и 7R в армию, то первый игрок в качестве комбинации может выбрать карту 7R и это будет самая старшая карта в игре, значит ход будет успешным для первого.
- Если масть G, то первому игроку надо будет выложить карту 4G на поле. Все остальные карты у него нечётные, а значит даже если он выложит ещё одну карту на поле, всё равно у него не наберётся комбинации даже из одной карты. Значит такой ход игрок сделать не может по правилам игры.
- Карт остальных мастей нет у первого игрока, значит он не может изменить масть поля на них.

Тогда первый игрок может сделать ход любым из первых двух способов, после чего второй игрок не сможет сделать ход и первый победит.

[А] 2. День выборов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

До выборов остался лишь один день. И ликёро-водочный завод, и военные, и казаки готовы голосовать за Игоря Владимировича. Однако сотрудникам «Как бы радио» этого мало, поэтому они послали отца Иннокентия провести агитационный молебен в поддержку Цаплина в разных церквях Приволжска.

Приволжск — это город квадратной формы, разделённый прямоугольной сеткой на квадратные районы, которые можно представить как клетки таблицы $n \times n$. В районе, находящемся в i -й строке в j -м столбце живёт $a_{i,j}$ человек. Между каждой парой соседних по стороне районов есть церковь, рассчитанная на прихожан этих двух районов (это означает, что к каждому району относится от 2 до 4 церквей).

В условиях ограниченного времени отец Иннокентий успеет провести агитационный молебен только в k церквях города. Теперь он хочет выбрать эти церкви так, чтобы как можно меньше людей не имели возможности прийти на агитационные молебны, то есть чтобы минимизировать общее количество жильцов районов, таких что во всех церквях, к ним относящихся, не проходили молебны. При этом отец Иннокентий не хочет проводить молебен в одном и том же районе несколько раз, даже если он проходит разных церквях, относящихся к району. Помогите отцу Иннокентию.

Формат входных данных

В первой строке даны 2 числа n и k ($2 \leq n \leq 2000$, $1 \leq k \leq 8$) — размер города и максимальное количество молебнов.

В следующих n строках дано по n чисел. В i -й строке на j -м месте дано число $a_{i,j}$ ($0 \leq a_{i,j} \leq 1000$) — количество жителей района, находящегося в i -й строке j -м столбце города в представлении города в виде таблицы.

Формат выходных данных

В единственной строке выведите одно число — минимальное общее население всех районов, таких что ни в одной относящейся к ним церкви не прошло агитационного молебна в поддержку Игоря Владимировича.

Примеры

стандартный ввод	стандартный вывод
3 1 2 7 6 9 5 1 4 3 8	31
4 2 1 2 4 0 4 0 5 4 0 3 5 1 1 0 4 1	17

Замечание

В первом примере молебен проходит в церкви, относящейся к районам с 9 и 5 жителями (2 строка 1 столбец и 2 строка 2 столбец).

Во втором примере первый молебен проходит в церкви, находящейся между районами в 3 столбце 1 строке и 3 столбце 2 строке, а второй молебен проходит в церкви, относящейся к районам в 3 столбце 3 строке и 3 столбце 4 строке. (Таким образом, во всех районах в третьем столбце прошёл молебен, а во всех остальных — нет).

Агитационный молебен:

А ты лети-лети над Волгой
Наш весёлый звонкий клич:
К нам приехал и надолго
Игорь Владимирович!
Аминь

Это наш счастливый случай,
Наш московский кандидат!
Игорь Цаплин самый лучший!
Остальные — ерунда!
Аминь

А кто простой российский парень
Не чечен и не москвич? —
Наш потомственный волжанин
Игорь Владимирович!
Аминь

Все ему родные тута,
Казачи и рыбаки.
Игорь Цаплин — это круто!
Остальные — дураки!
Аминь

А кто разгонит в небе тучи?
Кто всего может достичь? —
Наш великий и могучий
Игорь Владимирович!
Аминь

Аты-баты шли дебаты,
Мы пойдём на выборы.
Игорь Цаплин — терминатор!
Остальные — ...
Аминь

[A] 3. Osas Land

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

В никому не известной стране жил обладатель очень сложного имени (<https://algocode.ru/osas>) (далее Osas). Он захотел баллотироваться в президенты этой страны, но для этого сначала надо решить проблемы области, в которой он живёт, ведь там нет ни одной дороги!

Всего есть n сёл и m ещё не построенных дорог. i -я дорога будет соединять сёла v_i и u_i и обойдётся в c_i бублей. Надо построить какие-то из этих дорог так, чтобы из любого села можно было добраться по дорогам до любого села, а так же, чтобы между любыми двумя сёлами маршрут определялся однозначно (иначе надо будет платить за разработку навигатора).

Для этого Osas заранее выберет для каждой дороги её тип. Дороги первого типа имеют асфальтовое покрытие, а другие... лучше не знать. Оба покрытия стоят одинаково и не влияют на стоимость строительства дорог между сёлами. Но он хочет, чтобы была построена хотя бы одна дорога каждого типа, а так же чтобы строительство стоило как можно меньше.

На строительство дорог было выделено X бублей и Osas хочет потратить **ровно** столько на строительство дорог. Помогите ему посчитать количество способов назначить каждой дороге один из двух типов, чтобы минимальная стоимость строительства дорог, однозначно связывающих все города и содержащих оба типа покрытия, была равна X . Так как это число может быть очень большим выведите его по модулю $10^9 + 7$.

Формат входных данных

В первой строке вводятся два числа n и m ($1 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 3 \cdot 10^5$) — количество сёл и ещё не построенных дорог.

Во второй строке вводится число X ($1 \leq X \leq 2 \cdot 10^{14}$) — бюджет, выделенный на строительство дорог.

Далее следуют m строк, i -я из которых содержит три числа v_i, u_i, c_i ($1 \leq v_i, u_i \leq n, v_i \neq u_i, 1 \leq c_i \leq 10^9$) — номера сёл, которые соединяет i -я дорога, а так же стоимость строительства этой дороги.

Гарантируется, что между одной и той же парой сёл предлагается для строительства не больше одной дороги, а так же если построить все дороги, то от любого села можно будет добраться до любого другого по дорогам (но возможно не единственным способом).

Формат выходных данных

Выведите количество способов назначить каждой дороге один из двух типов, чтобы выполнялись условия выше, по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
4 6 12 1 4 8 3 2 6 3 4 9 4 2 5 1 2 4 3 1 3	48
6 10 33 2 1 12 1 6 7 3 1 12 2 5 9 3 2 6 2 4 17 5 1 3 1 4 12 4 3 5 6 5 10	60

[A] 4. Периоды...

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

Назовём строку s длины n хорошей, если её минимальный по длине период равен самой строке. Строка t является периодом строки s , если существует натуральное число $k > 1$, что s равна строке t , выписанной k раз.

Надо найти минимальное число подстрок, на которые можно разбить строку s так, чтобы каждая подстрока была хорошей и каждый символ принадлежал ровно одной подстроке.

Выведите количество подстрок в минимальном разбиении, а так же количество способов разбить строку таким образом на минимальное количество подстрок по модулю 998244353.

Строка t является подстрокой s , если t может быть получена из s удалением нескольких (возможно, ни одного или всех) символов из начала и нескольких (возможно, ни одного или всех) символов из конца.

Формат входных данных

В первой и единственной строке вводится строка s ($1 \leq |s| \leq 5 \cdot 10^5$).

Формат выходных данных

В первой строке выведите минимальное количество подстрок в разбиении.

Во второй строке выведите количество способов таким образом разбить строку на минимальное количество подстрок.

Примеры

стандартный ввод	стандартный вывод
bbb	3 1
abababab	2 4
abcabca	1 1

Замечание

В первом примере можно единственным образом разить строку на хорошие подстроки: $b|b|b$.

Во втором примере можно 4 способами разбить строку на две хороших подстроки: $a|bababab$, $aba|babab$, $ababa|bab$, $abababa|b$. Ответ не может быть 1, так как минимальный период строки $abababab$ это ab .

В третьем примере минимальный период строки s это сама строка s , поэтому ответ 1.

[A] 5. Запоминающийся путь

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

Родители отправляют ребенка в школу на автобусе. Всего в городе есть n остановок, соединённых $n - 1$ проездом так, что от любой остановки можно добраться до любой другой. Каждая остановка покрашена одним из 26 возможных цветов, обозначенных буквами латинского алфавита от «a» до «z».

Родители хотят выбрать две остановки, на одной из них построить свой дом, а на другой построить школу для ребёнка. При этом они хотят выбрать такие две остановки для школы и дома, что последовательность цветов остановок на кратчайшем пути от дома до школы должна совпадать с последовательностью цветов остановок на кратчайшем пути от школы до дома. Найдите длину самого большого пути от школы до дома, соблюдая эти условия. Дом и школа могут находиться на любых остановках.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 50\,000$) — количество остановок в городе.

Вторая строка содержит одну строку s из n строчных латинских букв — цвета остановок.

В i -й из следующих $n - 1$ строк задано два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$), обозначающих очередную дорогу на схеме городского транспорта.

Формат выходных данных

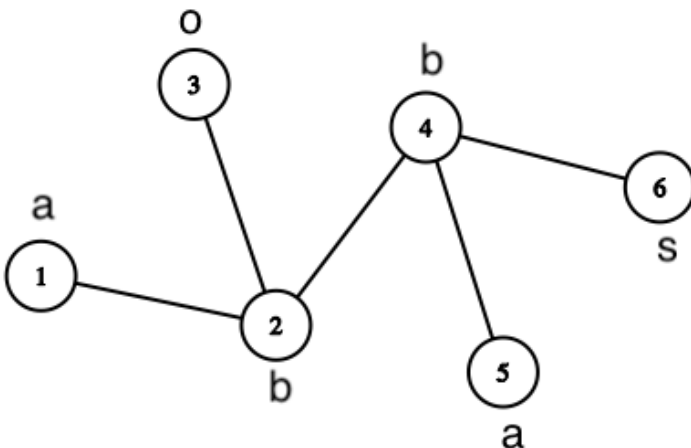
Выведите единственное число — максимальную длину пути между двумя остановками, такую что последовательность цветов остановок на прямом и обратном движении по пути совпадают.

Пример

стандартный ввод	стандартный вывод
6 abobas 1 2 2 3 5 4 6 4 2 4	4

Замечание

Последовательность цветов на пути 1 – 2 – 4 – 5 совпадает с последовательностью цветов на пути 5 – 4 – 2 – 1. Это самый большой путь, удовлетворяющий условию, а его длина равна 4.



[A] 6. Простая задача

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

Есть гирлянда, состоящая из бесконечного числа лампочек с номерами $1, 2, 3, \dots$

Скоро начнутся занятия в Tinkoff Generation и вы хотите, чтобы в честь такого события все лампочки были включены. К счастью только n лампочек выключены и с гирляндой можно выполнять простые операции:

1. Выбрать простое число p строго больше 2.
2. Выбрать натуральное число l .
3. Изменить состояние лампочек с номерами $l, l+1, \dots, l+p-1$ (то есть выключить включённые лампочки и включить выключенные).

Вы хотите включить все лампочки как можно быстрее, поэтому вас интересует минимальное количество операций, за которое можно этого добиться.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 100$) — количество выключенных лампочек.

Вторая строка содержит n натуральных чисел x_1, \dots, x_n ($1 \leq x_1 < \dots < x_n \leq 10^7$) — номера выключенных лампочек.

Формат выходных данных

Выведите минимальное количество операций, за которое можно включить все лампочки.

Примеры

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

Замечание

В первом примере можно изменить состояние лампочек $3, 4, 5, 6, 7$, выбрав $p = 5$ и $l = 3$. Тогда выключены будут только лампочки $5, 6, 7$. Далее можно изменить состояние этих лампочек, выбрав $p = 3$ и $l = 5$.

Во втором примере можно включить все лампочки за 3 операции: $(p = 3, l = 2)$, $(p = 3, l = 5)$, $(p = 3, l = 8)$.