

Problem A. Абстракционист и треугольники

Input file:

стандартный ввод

Output file:

стандартный вывод

Time limit:

1 секунда

Memory limit:

1536 мебибайт

Это задача с двойным запуском

Абстракционист Калевич хочет переслать несколько неравносторонних треугольников на организуемую в офисе МТС выставку треугольников. Но так как почта Берляндии берёт оплату за периметр треугольника, Калевич хочет сэкономить.

Он попросил Вас закодировать треугольник ненулевой площади с целыми сторонами и периметром p , никакие две стороны которого не равны, треугольником ненулевой площади и периметром $p - 6$, и затем по закодированному треугольнику восстановить исходный.

Input

Первая строка входных файлов содержит режим работы: `encode`, если требуется закодировать треугольники или `decode`, если требуется восстановить исходные. Вторая строка содержит одно целое число t ($1 \leq t \leq 25\,000$) — количество тестовых примеров.

Далее следуют тестовые примеры по одному на строку.

В режиме кодирования каждый тестовый пример содержит три целых числа a , b и c — стороны исходного треугольника. Гарантируется, что $3 \leq a + b + c \leq 1000$, что a , b и c попарно не равны и что треугольник со сторонами a , b и c существует и имеет ненулевую площадь.

В режиме декодирования каждый тестовый пример содержит три целых числа a' , b' и c' — переставленные в каком-то порядке стороны, выведенные вашей программой в режиме кодирования.

Output

В режиме кодирования выведите для каждого тестового примера в произвольном порядке три целых числа a' , b' , c' , такие, что треугольник со сторонами a' , b' , c' существует и имеет ненулевую площадь, и $a' + b' + c' = a + b + c - 6$.

В режиме декодирования для каждого тестового примера выведите три стороны исходного треугольника. Порядок вывода сторон внутри одного тестового примера произвольный.

Examples

стандартный ввод	стандартный вывод
encode 2 20 24 11 25 36 49	18 18 13 25 30 49
decode 2 18 13 18 25 49 30	11 20 24 25 36 49

Problem B. Больше MTS-строк!

Input file: *стандартный ввод*
 Output file: *стандартный вывод*
 Time limit: 3 секунды
 Memory limit: 1536 мебибайт

Назовём строку, составленную из заглавных букв латинского алфавита, *MTS-строкой*, если в этой строке можно удалить какое-то (возможно, нулевое) количество букв так, чтобы получилась строка MTS.
 Назовём строку *k*-устойчивой MTS-строкой, если при любом разбиении этой строки на *k* непрерывных подстрок хотя бы одна строка будет MTS-строкой.
 Для заданных *n* и *k* найдите остаток от деления количества *k*-устойчивых MTS-строк длины *n* на 998 244 353.

Input

Первая строка входных данных содержит два целых числа *n* и *k* ($1 \leq n, k \leq 3 \cdot 10^4$).

Output

Выведите одно целое число — остаток от деления количества *k*-устойчивых MTS-строк длины *n* на 998 244 353.

Examples

стандартный ввод	стандартный вывод
4 1	101
3 2	0

Problem C. Весёлая реклама 2

Input file:	стандартный ввод
Output file:	стандартный вывод
Time limit:	3 секунды
Memory limit:	1024 мегабайта

Это задача с двойным запуском. Первый запуск является интерактивным

Недавно вышла новая линейка тарифов МТС, при этом каждый тариф имеет не только своё название, но и свой слоган! Тарифов в линейке очень много, так что даже с помощью слоганов перечислить клиенту все тарифы будет сложно.

Сотрудники отдела рекламы придумали неожиданный вариант решения проблемы — было предложено составить из слоганов стихотворные тексты.

Тексты были заказаны специалисту по стихотворной рекламе. Однако он потребовал уточнения технического задания — сообщить схему рифмовки.

Схема рифмовки строится следующим образом. Пусть в одном куплете рекламы ℓ строк. Тогда схема рифмовки — строка из ℓ заглавных латинских букв, обладающая следующим свойством:

- Первая буква строки всегда равна А и соответствует первой строке.
- Если строка рифмуется с уже существующей, она обозначается такой же буквой.
- Если строка не рифмуется ни с одной из существующих, она обозначается лексикографически наименьшей неиспользованной буквой.

Например, в четверостишии, где рифмуется первая строка с третьей, а вторая — с четвёртой, схема рифмовки будет АВАВ. Если все строки рифмуются между собой — схема будет АААА. А вот, например, в японском танка из 5 не рифмующихся строк схема рифмовки будет АВСDE.

Проблема была в том, что у специалиста стоял очень старый мессенджер «Морзянка» с максимальным уровнем защиты, который в качестве сообщений принимает только наборы из не более, чем 50 попарно взаимно простых целых чисел, больших единицы и меньших 10^{18} .

Сеанс связи состоит из нескольких таких сообщений. При этом наименьшее общее кратное всех передаваемых за сеанс связи чисел не должно превосходить 10^{18} .

Вам даётся набор схем рифмовок для разных ℓ , требуется написать программу, которая передаёт строки через мессенджер по одному сообщению на строку за один сеанс связи, а затем декодирует полученную информацию, восстанавливая схемы рифмовок. При первом запуске вы получаете схемы и эмулируете сеанс связи, отправляя закодированные сообщения по очереди. При втором запуске вы получаете информацию, переданную при первом запуске (с точностью до перестановки чисел внутри одного сообщения), и должны восстановить переданные схемы рифмовки.

Interaction Protocol

При работе в режиме передачи первая строка входных данных содержит слово **encode**. Далее сообщается количество t схем рифмовки, которые надо передать в текущем сеансе связи ($1 \leq t \leq 2 \cdot 10^5$).

Далее t раз взаимодействие происходит по следующему сценарию: вы получаете строку длиной не менее двух и не более 15 символов, состоящую из заглавных латинских букв и задающих схему рифмовки, после чего генерируете сообщение в виде k попарно взаимно простых целых чисел a_i , больших единицы ($1 \leq k \leq 50$), и выводите в первой строке k , а во второй — требуемые k чисел в правильном порядке. Не забывайте после вывода k и после вывода строки с a_i выводить команду перевода строки и сбрасывать буфер вывода вызовом функции **flush** используемого Вами языка программирования.

Обратите внимание, что ограничение на НОК всех передаваемых чисел работает на весь сеанс (то есть на все числа, передаваемые во всех сообщениях).

Input

При работе в режиме приёма первая строка входных данных содержит слово **decode**. Вторая строка содержит количество сообщений t . Далее следуют сами сообщения в том же формате, в котором они были переданы в режиме передачи — на первой строке количество сообщений k , далее k целых чисел a_i . При этом порядок чисел a_i внутри строки может быть изменён.



Output

Для каждого из t сообщений выведите схему рифмовки, которая была закодирована этим сообщением.

Examples

стандартный ввод	стандартный вывод
encode 2 AA AB	 2 3 4 1 5
decode 2 2 4 3 1 5	AA AB

Problem D. Граф и запросы

Input file:

стандартный ввод

Output file:

стандартный вывод

Time limit:

2 секунды

Memory limit:

1024 мегабайта

Дан неориентированный граф G , состоящий из n вершин и m рёбер. У каждой вершины есть значение x_v . Назовем *характеристикой* графа величину $c(G) = \min_{u,v} |x_u - x_v|$ по всем таким парам вершин $u \neq v$, что из u в v существует путь. Если в графе нет рёбер вообще, положим $c(G) = -1$. Вам необходимо найти характеристику графа в самом начале, а также после выполнения каждого из q запросов. В i -ом запросе вам нужно удалить из графа ребро с номером nit_i . Обратите внимание, все удаления в графе сохраняются, в i -ом запросе граф без рёбер $nit_1, nit_2, \dots, nit_i$.

Input

В первой строке вам даны числа $1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$ и $1 \leq q \leq 2 \cdot 10^5$ – количество вершин графа, рёбер графа и количество запросов. В следующей строке вам дано n целых чисел $-10^9 \leq x_i \leq 10^9$ – значения вершин. В следующих m строках заданы рёбра графа – по два числа в каждой строке $1 \leq u_i \neq v_i \leq n$, все рёбра различны. В следующей строке дано q чисел – номера рёбер $1 \leq nit_i \leq m$ в порядке их удаления из графа.

Output

Выведите $q + 1$ чисел – характеристику графа до начала выполнения запросов и после выполнения каждого из них.

Example

стандартный ввод	стандартный вывод
<div> <div>8 10 10</div> <div>1 2 6 3 14 8 1 15</div> <div>7 8</div> <div>3 7</div> <div>3 6</div> <div>2 3</div> <div>1 8</div> <div>4 3</div> <div>6 4</div> <div>2 5</div> <div>1 3</div> <div>5 6</div> <div>5 2 7 3 4 6 9 10 8 1</div> </div>	<div> <div>0 0 1 1 1 2 5 6 12 14 -1</div> </div>

Problem E. Дорожные работы

Input file: *стандартный ввод*
Output file: *стандартный вывод*
Time limit: 1 секунда
Memory limit: 1024 мегабайта

В новом микрорайоне проложили кабельный интернет от МТС, соединив каждую пару домов сегментом оптоволоконного кабеля. Всего в микрорайоне n домов, занумерованных последовательными целыми числами от 1 до n . Месячная стоимость обслуживания сегмента кабеля между домами u и v равна $\min(u, v)$.

Во время планового ремонта дорог в микрорайоне было повреждено k сегментов кабеля, при этом связь между любыми двумя домами через один или несколько кабелей всё ещё была возможна. В итоге разбирательства дорожный трест был оштрафован на сумму, равную месячной стоимости обслуживания такого набора оставшихся неповреждёнными сегментов, который соединял бы все дома микрорайона. Разумеется, трест хочет выбрать набор так, чтобы заплатить как можно меньшую сумму. Ваша задача — найти эту сумму.

Input

В первой строке вам даны числа $1 \leq n \leq 10^5$ — количество домов в микрорайоне и $0 \leq k \leq \min(\frac{n \cdot (n-1)}{2}, 3 \cdot 10^5)$ — количество уничтоженных сегментов.

В следующих k строках содержатся описания уничтоженных сегментов — по два различных целых числа от 1 до n включительно. Гарантируется, что среди уничтоженных сегментов нет одинаковых и что с помощью оставшихся неповреждёнными сегментов можно объединить в сеть все дома микрорайона.

Output

Выведите ответ на задачу — минимальную сумму, которую трест заплатит в качестве штрафа.

Example

стандартный ввод	стандартный вывод
7 11 1 2 1 3 1 4 1 5 1 6 2 3 2 4 2 5 4 6 3 7 3 6	15

Problem F. Ежегодный фестиваль

Input file: *стандартный ввод*
 Output file: *стандартный вывод*
 Time limit: 1 секунда
 Memory limit: 1024 мегабайта

Ежегодный музыкальный фестиваль проходит на стадионе размером $r \times c$. Стадион разбит на единичные квадраты.

Для обеспечения участников фестиваля качественным мобильным интернетом специалисты компании МТС хотят поставить в некоторые из квадратов усилители сигнала, не более, чем по одному усилителю сигнала на единичный квадрат. Согласно проведённым исследованиям, покрытие будет стабильным в случае, когда количество усилителей в каждой строке и каждом столбце нечётно.

Всего для этих целей планируется установить k усилителей. Сколькими способами можно расставить все усилители так, чтобы покрытие было стабильным? Две расстановки считаются различными, если существует какой-то единичный квадрат, в котором для одной расстановки есть усилитель, а для другой нет.

Так как ответ может быть очень большим, выведите остаток от его деления на 998 244 353.

Input

Входные данные содержат три целых числа r , c и k ($1 \leq r, c \leq 5 \cdot 10^5$, $r \times c \leq 5 \cdot 10^5$, $1 \leq k \leq r \times c$).

Output

Выведите одно целое число — остаток от деления ответа на 998 244 353.

Examples

стандартный ввод	стандартный вывод
2 2 2	2
3 2 2	0