Задача A. Аккуратная табличка

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

С чего начинается проведение очередного сезона RuCode? С составления задач? С выбора столицы? Не совсем. Ещё до всего этого нужно сгенерировать пары «логин-пароль» для пользователей.

В системе, в которой происходит генерация логинов, в запросе можно указать префикс логина, например, `rucode24final-` и два числа s и n ($1 \leqslant s \leqslant 10^6$, $1 \leqslant n \leqslant 500$). Будут сгенерированы логины с n подряд идущими номерами, начиная с s-го.

Система генерирует логины с заданным префиксом и всеми номерами из диапазона. Например, если попросили сгенерировать логины с 8 по 10 ($s = 8$ и $n = 3$), будет выдано `rucode24final-8`, `rucode24final-9`, `rucode24final-10`.

Уже на разобранном примере видно, что логины могут иметь различную длину, поэтому табличка с ними выглядит неаккуратно. Для того, чтобы избежать такого, можно «добивать» префикс ведущими нулями, то есть для гернерации логинов с номерами с первого по десятый придётся использовать два запроса: один с префиксом «`rucode24final-0`», $s = 1$ и $n = 9$, а второй с префиксом «`rucode24final-`», $s = 10$ и $n = 1$. Тогда получится 10 логинов одинаковой длины с `rucode24final-01` по `rucode24final-10` включительно.

Сколько запросов потребуется, чтобы сгенерировать n подряд идущих, начиная с единицы, логинов равной длины, следуя правилам, описанным выше? Напоминаем, что система не позволяет за один запрос сгенерировать более 500 логинов.

Формат входных данных

Первая строка входных данных содержит одно целое число n — количество логинов, которые должны быть сгенерированы ($1 \leqslant n \leqslant 20\,000$).

Формат выходных данных

Выведите одно целое число — количество запросов на генерацию пароля.

Примеры

стандартный ввод	стандартный вывод
3	1
25	2

Задача B. Бордеры

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

У вас есть две строки a и b длин n и m соответственно.

Строка t называется *бордером* (или гранью) строки s, если t является одновременно как префиксом, так и суффиксом s. Строка t называется префиксом s, если начало s(строка, представляющая собой конкатенацию первых $|t|$ символов s) совпадает с t. Аналогично, строка t называется суффиксом t, если конец s(строка, полученная конкатенацией последних $|t|$ символов s) совпадает с t. Здесь $|t|$ обозначает длину t.

Строка w называется *общим бордером* строк a и b, если w – бордер как a, так и b.

Строка w называется *наименьшим непустым общим бордером* строк a и b, если она имеет наименьшую длину среди всех общих бордеров строк a и b и она непуста(её длина больше нуля).

Обозначим как $LCB(s,t)$ наименьший непустой общий бордер строк s и t, либо пустая строка, если у строк s и t нет общего бордера.

За $s[l..r]$ обозначим подстроку строки s, начинающуюся в l и заканчивающуюся в r(подстрока – последовательность подряд идущих символов).

Обозначим как $res_i = \sum_{l=1}^{n} \sum_{r=l}^{n} |LCB(b[1..i], a[l..r])|$. То есть это сумма длин наименьших непустых общих бордеров для i-го префикса b и всех подстрок a.

Ваша задача – посчитать набор значений res.

Формат входных данных

В первой строке вам дано количество тестовых примеров q ($1 \leqslant q \leqslant 5 \cdot 10^5$).

В следующих строках задаются сами тестовые примеры. Каждый тестовый пример состоит из двух строк a и b, разделенных переводом строк. Гарантируется, что суммарная длина всех строк a не превосходит $5 \cdot 10^5$ и суммарная длина всех строк b не превосходит $5 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите m чисел(где m – длина строки b) – значения res для соответствующего тестового примера.

Пример

стандартный ввод	стандартный вывод
3 aba bab baceca acca ettaotta ttettay	1 2 1 3 2 0 3 10 10 0 10 10 0 0

Задача C. Внимание к деталям!

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

В мастерской по изготовлению римских чисел для нужд фестиваля RuCode есть три вида деталей: вертикальная палочка, наклонная палочка и полукруг. Каждую деталь можно отразить относительно вертикальной оси.

Таким образом, из этих деталей цифры делаются следующим образом:

- I — одна вертикальная палочка;
- V — две наклонных палочки;
- X — две наклонных палочки;
- L — из данного набора не делается;
- C — один полукруг;
- D — вертикальная палочка плюс полукруг;
- M — две вертикальные и две наклонные палочки

В мастерскую поступили a вертикальных палочек, b наклонных палочек и c полукругов. Требуется собрать из них римские числа так, чтобы каждая деталь была задействована ровно в одной цифре одного числа, а общее количество римских чисел было минимальным.

Напоминаем, что современные римские числа записываются следующим образом (в соответствии с таблицей из Википедии):

Значение разряда	Тысячи	Сотни	Десятки	Единицы
1	M	C	X	I
2	MM	CC	XX	II
3	MMM	CCC	XXX	III
4		CD	XL	IV
5		D	L	V
6		DC	LX	VI
7		DCC	LXX	VII
8		DCCC	LXXX	VIII
9		CM	XC	IX

Правила построения:

- Числа 4, 9, 40, 90, 400 and 900 записываются в реверсивной нотации, где первый символ вычитается из второго (например, для 40 (`XL`) '`X`' (10) вычитается из '`L`' (50)). Это **единственные** места, где реверсивная нотация используется.
- Число, содержащее несколько десятичных цифр, строится дописыванием римского эквивалента каждой цифры от старшего разряда к младшему.
- Если в десятичном разряде стоит 0, никаких цифр в этом разряде в римском представлении не пишется.
- Наибольшее число, которое может быть представлено в римской системе счисления — это число 3,999 (MMMCMXCIX).

Формат входных данных

Первая строка входных данных содержит одно целое число t — количество тестовых примеров ($1 \leqslant t \leqslant 30\,000$).

Каждая из последующих n строк содержит по три целых числа a_i, b_i и c_i ($0 \leqslant a_i, b_i, c_i \leqslant 100$) — количество деталей, поступивших в мастерскую в i-м тестовом примере.

Формат выходных данных

Если при любой сборке римских чисел какая-то деталь останется неиспользованной, выведите для соответствующего тестового примера -1. Иначе выведите одно целое число — минимальное количество римских чисел, которое можно собрать из соответствующего набора деталей так, чтобы все детали были использованы по одному разу.

Пример

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

Задача D. Готовим интенсив...

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

Это интерактивная задача

Алиса и Боб ведут занятия в дивизионе G Рукода. Перед первой лекцией им надо добавить n задач в контест интенсива. Изначально контест пустой. Задачи должны быть пронумерованы n подряд идущими заглавными латинскими буквами, начиная с `A`.

Задачи добавляются по очереди, первой добавляет Алиса. При добавлении задачи можно или выбрать, какая буква будет соответствовать задаче, или положиться на подсказку системы.

Подсказка системы работает следующим образом: если в контесте уже есть x задач, будет предложено выбрать букву $x + 1$ и сделать один клик мышкой. Разумеется, может оказаться, что такая буква уже используется (например, перед добавлением контест состоит из задач C, D и F; так как задач 3, система предложит в качестве подсказки D, но задача D уже есть в системе), тогда использовать подсказку не получится и придётся выбирать другую букву с помощью клавиатуры.

Алиса заметила, что Боб ленится нажимать кнопки на клавиатуре и по возможности хочет, чтобы система выбирала за него (то есть если возможность принять букву, предложенную системой, есть, Боб всегда её принимает).

В педагогических целях Алиса решила провести следующий эксперимент: она хочет добавлять задачи в таком порядке, чтобы ровно в k случаях у Боба не получилось использовать подсказку. Сможете ли вы придумать такую последовательность действий?

Протокол взаимодействия

Взаимодействие начинает программа жюри, сообщая два целых числа n ($3 \leqslant n \leqslant 15$) — количество задач в контесте интенсива и k ($0 \leqslant k \leqslant \lfloor n/2 \rfloor$) — количество задач, буквы для которых Боб в итоге должен выбрать сам.

Далее ваша программа, которая действует за Алису, должна выбрать букву задачи и сообщить её программе жюри, выводя одну заглавную букву от `A` до n-й подряд, начиная с `A`, буквы алфавита. Если такая буква задачи уже есть в контесте, вы проиграли и получите вердикт `Wrong Answer`. Если такой буквы задачи нет, задача добавляется в контест. Если добавлены ещё не все задачи, программа жюри делает ход за Боба в соответствии с описанной в условии тактикой (то есть если есть возможность использовать подсказку системы, то подсказка используется, иначе выбирается какая-то буква произвольным образом).

После того, как все задачи добавлены, программа жюри подсчитывает количество раз, когда подсказка системы не была использована. Если это количество не равно k, вы также получите вердикт `Wrong Answer`.

Пример

стандартный ввод	стандартный вывод
3 0 B	 A C

Замечание

Не забывайте после выбора очередной буквы задачи выводить символ перевода строки, а также сбрасывать буфер ввода-вывода вызовом функции `flush` используемого вами языка программирования.

Задача E. Делимость и займы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

МТС-банк планирует открыть новую беспроцентную кредитную программу для одарённых студентов «Делим и делимся». В рамках кредитной программы студент получает кредит в n рублей и может вернуть его в течение k месяцев, в i-й месяц возвращая x_i рублей, то есть $x_1 + x_2 + ... + x_k = n$.

При этом числа x_i могут быть произвольными целыми положительными числами, удовлетворяющими следующим ограничениям.

- $x_1 \leqslant x_2 \leqslant ... \leqslant x_k$
- n делится на x_k
- $n - x_k$ делится на x_{k-1}
- $n - x_k - x_{k-1}$ делится на x_{k-2} ...
- $n - x_k - x_{k-1} - ... - x_3$ делится на x_2
- $n - x_k - x_{k-1} - ... - x_3 - x_2$ делится на x_1

Студентка Катя заинтересовалась, а сколько существует возможных различных наборов x_i, удовлетворяющих этим ограничениям (два набора x_i и x'_i считаются различными, если для какого-то i $x_i \neq x'_i$). Так как ответ может быть очень большим, выведите остаток от его деления на 998 244 353.

Формат входных данных

Входные данные содержат два целых числа n и k ($1 \leqslant n \leqslant 10^{18}$, $1 \leqslant k \leqslant 10$).

Формат выходных данных

Выведите остаток от деления количества различных наборов x_i, задающих корректный порядок возврата кредита, на 998 244 353.

Пример

стандартный ввод	стандартный вывод
12 5	5

Замечание

В первом примере существуют следующие решения:
$12 = 1 + 1 + 1 + 3 + 6$
$12 = 1 + 1 + 2 + 2 + 6$
$12 = 1 + 1 + 2 + 4 + 4$
$12 = 2 + 2 + 2 + 2 + 4$
$12 = 2 + 2 + 2 + 3 + 3$

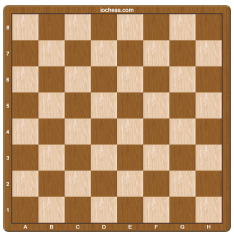
Задача F. Естественная видимость

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

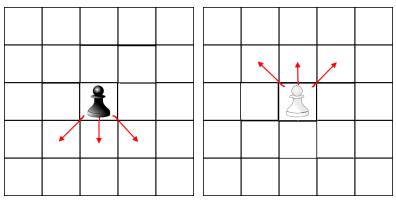
Шахматы с туманом войны («dark chess») отличаются от обычных шахмат тем, что в них игрок видит не всё поле, а только те клетки, до которых могут дойти его фигуры. Вам дана расстановка фигур, требуется определить, что видит каждый игрок.

В данной задаче действуют упрощённые правила шахмат (в том числе расстановки фигур), отличающиеся от реальных правил игры:

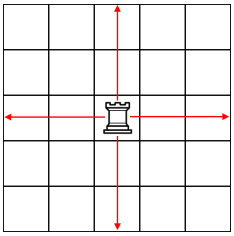
- Игра идёт на доске 8×8, поля которой раскрашены в белые и чёрные цвета так, чтобы никакие два поля, имеющие общую границу, не имели один и тот же цвет. Левое нижнее поле доски является чёрным.



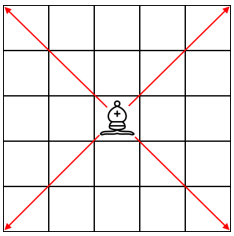
- Поле может быть либо свободным, либо занятым какой-то одной фигурой. Фигура имеет цвет (чёрная или белая) и тип (король, ферзь, конь, слон, ладья, пешка).
- У каждой стороны (белых и чёрных) на доске обязан присутствовать ровно один король.
- С каждой стороны может быть от нуля до восьми пешек, от нуля до двух ладей, от нуля до двух коней, от нуля до двух слонов и от нуля до одного ферзя. Если у игрока есть два слона, один из них стоит на чёрном поле, а второй — на белом.
- Пешки не могут располагаться на самой нижней и самой верхней горизонталях.
- Клетки, занимаемые фигурами данного цвета, всегда являются видимыми для игрока, который играет этим цветом.
- Фигуры не могут выходить за пределы доски.
- Чёрная пешка может пойти на одно поле вниз, белая — на одно поле вверх. Пешка может брать фигуру, только если та расположена в квадрате, имеющим с текущим ровно одну общую точку и расположенном в направлении движения пешки (то есть «по диагонали вперёд». Пешка видит как то поле, на которое она может пойти, так и те поля, на которых она могла бы брать фигуру, вне зависимости от наличия там фигуры. Никаких других ходов пешек, которые встречаются в обычных шахматах, в данной задаче нет.



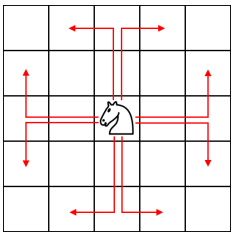
- Ладья может пойти на любое свободное или занятое фигурой другого цвета поле в той же горизонтали или вертикали, на которой она стоит, при условии, что между ней и этим полем отсутствует другая фигура.



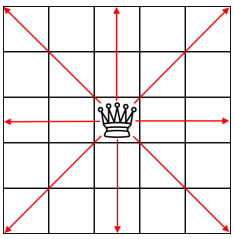
- Слон может пойти на любое свободное или занятое фигурой другого цвета поле в любом из четырёх диагональных направлений, при условии, что между ним и этим полем отсутствует другая фигура.



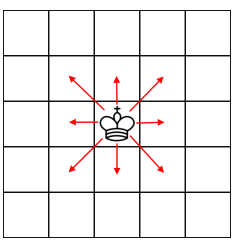
- Конь может пойти на любое свободное или занятое фигурой другого цвета поле доски, до которого он может добраться, переместившись на две клетки в одном направлении и на одну клетку в перпендикулярном.



- Ферзь может пойти на любое из полей, на которое могли бы пойти слон или ладья того же игрока с того же самого поля, на котором стоит ферзь.



- Король может пойти на любое свободное или занятое фигурой другого цвета поле, имеющее с текущим хотя бы одну общую точку (то есть на любое из восьми соседних полей).



- Множество полей, видимых фигурой, совпадает с множеством полей, на которое она может пойти. Множество полей, видимых игроком, есть объединение множества полей, видимых фигурами соответствующего цвета.

Никакие другие правила (рокировка, взятие на проходе и так далее) в данной задаче не рассматриваются и на видимость доски не влияют. Ограничений на взаимное расположение фигур также нет (в частности, два короля и в реальных dark chess вполне могут стоять рядом)

Формат входных данных

Входные данные состоят из 8 строк по 8 символов в каждой. Если символ равен `.`, то соответствующее поле свободно. Если символ — строчная буква, то фигура белая. Буква `k` соответствует королю, буква `r` — ладье, буква `q` — ферзю, буква `n` — коню, буква `b` — слону и буква `p` — пешке.

Если символ – заглавная буква, то фигура чёрная. Буква `K` соответствует королю, буква `R` — ладье, буква `Q` — ферзю, буква `N` — коню, буква `B` — слону и буква `P` — пешке.

Формат выходных данных

Выведите два блока по 8 строк из 8 символов в каждой: первый блок соответствует видимости поля с точки зрения белых (поля, которые белые не видят, должны быть закрыты знаками `?`, а на остальных полях должны располагаться те же фигуры, что и во входном файле), второй блок — видимости с точки зрения чёрных в аналогичном формате.

Примеры

стандартный ввод	стандартный вывод
RNBKQBNR PPPPPPPP .... pppppppp rnbkqbnr	???????? ???????? ???????? ???????? ???????? .... pppppppp rnbkqbnr RNBKQBNR PPPPPPPP .... ???????? ???????? ???????? ???????? ????????
.r...... P...q.p. ....n..P P...b..P .kRp.... ..P..... .Kpp.... ..B.....	.r...... P...q.p? ?.?.n.?? P...b?.? .kRp?.?. ..P..?.? ??pp???. ???????? ??.????? P?.????? ...????P P?.???.P .kRp??.. ..P????? .Kpp???? ..B?????

Задача G. Ё322

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

Это интерактивная задача

Для крупного телекоммуникационного оператора жизненно важным является вопрос диверсификации используемого оборудования с сохранением унификации программного обеспечения. Поэтому инженеры компании МТС решили исследовать вопрос о переносимости системного программного обеспечения между принципиально разными типами процессоров.

Планируется разработка модуля автоадаптации с кодовым наименованием «Ё322», который будет распознавать архитектуру микропроцессора и генерировать для неё эффективный код. Ваша задача в рамках этой разработки — написать модуль, который отличает два типа организации потоков ввода-вывода — стек и очередь.

Дана структура данных p, представляющая собой или стек, или очередь. Структура может быть пустой или содержать некоторое количество (не более 297) элементов.

Ваш модуль получает эксклюзивный доступ к структуре и может задать до 300 запросов одного из двух видов `put` x — добавить число x (целое число от 1 до 1000) в структуру данных или `get` — получить значение из головы очереди (вершины стека) и удалить соответствующий элемент. Если структура пуста, вы получаете специальное сообщение. После 300 запросов (или ранее) вы должны ответить, стек это или очередь.

Протокол взаимодействия

Взаимодействие начинает Ваша программа, выводя одну из команд. Если вы планируете поместить элемент x в структуру данных, выведите команду `put` x ($1 \leqslant x \leqslant 1000$). Если вы планируете снять элемент и узнать его значение, выведите команду `get`. Программа жюри выведет одно целое число — значение взятого элемента или -1, если структура пуста. Гарантируется, что изначально структура содержит не более 297 элементов.

Как только вы готовы определить, какая именно структура данных загадана, выведите `stack`, если это стек, и `queue`, если это очередь. Это действие не учитывается при подсчёте количества запросов.

Обратите внимание, что интерактор является адаптивным, то есть финальный ответ достраивается по ходу взаимодействия (но всегда удовлетворяет всей имеющейся информации и условиям задачи).

Пример

стандартный ввод	стандартный вывод
3	put 3 put 4 get queue

Задача H. Журналистское исследование

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

Город M. состоит из n площадей, некоторые из этих площадей соединены трамвайными линиями с двусторонним движением, при этом любая пара площадей соединена не более, чем одной такой линией.

Два журналиста, специализирующиеся на вопросах урбанистики, решили провести следующий эксперимент: из всех возможных пар различных площадей случайно и равновероятно выбирается пара (A, B), после чего проверяется, есть ли трамвайный маршрут (возможно, с пересадками) между площадями A и B.

По заданной карте трамвайных линий найдите вероятность того, что такой маршрут найдётся.

Формат входных данных

Первая строка входных данных содержит два целых числа n и k — количество площадей и количество трамвайных линий в городе M. ($2 \leqslant n \leqslant 10^5$, $0 \leqslant k \leqslant \min(n(n-1)/2, 3 \cdot 10^5)$). Каждая из последующих k строк содержит по два целых числа a_i и b_i — номера площадей, которые соединяет очередная трамвайная линия ($1 \leqslant a_i, b_i \leqslant n$, $a_i \neq b_i$). Гарантируется, что одна пара площадей соединена напрямую не более, чем одной трамвайной линией (то есть если во входе есть пара (x, y), то она встречается ровно один раз, а пара (y, x) вообще не встречается).

Формат выходных данных

Можно доказать, что вероятность представляется в виде p/q, где $p \geqslant 0$, $q > 0$ и наибольший общий делитель p и q равен единице. Выведите соответствующие значения p и q через пробел.

Примеры

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

Задача I. Залипающие кнопки

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

Возможно вы замечали, что в домофонах некоторые кнопки могут работать хуже других. Например, потому что коды большинства квартир содержат цифру 1.

Компания МТС в рамках развития концепции «умного дома» устанавливает модифицированные домофоны, одним из преимуществ которых является защита от подобных спецэффектов.

Известно, что каждый код — это пятизначное число, состоящее из цифр от 0 до 9. Система назначает коды домофона для n квартир. Чтобы избежать проблемы залипания кнопок, необходимо, чтобы разность суммарных количеств использования любых двух цифр не превосходила 1. Формально, для любых двух цифр i и j $(0 \leqslant i, j \leqslant 9)$ должно выполняться $|cnt(i) - cnt(j)| \leqslant 1$, где $cnt(x)$ — суммарное количество использований цифры x по всем кодам.

Например, для множества кодов $\{11111,\ 12346,\ 12347\}$ и цифры 1 $cnt(1) = 7$.

Ваша задача — сгенерировать n различных пятизначных кодов, соблюдая это условие.

Формат входных данных

Единственная строка входных данных содержит одно целое число n $(1 \leqslant n \leqslant 10^5)$ — число квартир, для которых требуется назначить коды домофона.

Формат выходных данных

Выведите n уникальных пятизначных чисел такие, чтобы разность суммарных количеств использования любых двух цифр по всем числам не превосходила 1, по одному числу на строку. Если ответов несколько, выведите любой.

Пример

стандартный ввод	стандартный вывод
10	00000 11111 22222 33333 44444 55555 66666 77777 88888 99999

Задача J. «И» побитовое

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Рассмотрим массив из n неотрицательных чисел a_1, a_2, …, a_n. Обозначим за $f(x)$ минимальное число вида 2^k, $k \geqslant 0$, которое строго больше x.

За одно действие можно выбрать некоторое $1 \leqslant i \leqslant n$ и заменить a_i на $f(a_i) - 1 - a_i$.

Требуется максимизировать значение побитового «И» элементов массива, использовав какое-то количество действий (или не использовав ни одного действия). Какого максимального значения Крош хочет, чтобы побитовое «И» полученных элементов массива было как можно больше. Помогите ему понять, какого максимального его значения можно достичь.

Формат входных данных

В первой строке вам дано число $1 \leqslant n \leqslant 10^5$. В следующей строке вам даны n чисел — элементы массива. i-е из этих чисел, содержит элемент a_i, $0 \leqslant a_i \leqslant 10^9$.

Формат выходных данных

Выведите одно целое число — максимальное значение побитового «И», которого можно добиться за какое-то конечное (возможно, нулевое) количество действий.

Пример

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

Замечание

Побитовое «И» – битовая операция, которая выполняется над двумя числами следующим образом: числа рассматриваются в двоичной системе счисления, а затем к каждой цифре в отдельности применяется операция логического «И». Логическое «И» равно единице тогда и только тогда, когда обе цифры, над которыми мы его выполняем, равны единице, и нулю в противном случае. В языке C++ операция побитового «И» выполняется с помощью «&».

Например, 17_{10} *and* $13_{10} = 10001_2$ *and* $01101_2 = 00001_2 = 1_{10}$

Задача K. Йода принимает экзамен

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

Это интерактивная задача

...В далёкой-далёкой галактике принято выделять n классов звёзд. Все звёзды одного класса — это звёзды, масса которых в миллиграммах находится в определённом интервале. Каждый класс обозначен какой-то строчной латинской буквой, разные классы обозначены разными буквами (буквы не обязательно идут подряд).

Таблица классов звёзд тем самым является набором из n пар «буква — максимальная масса», отсортированных в порядке возрастания максимальной массы. Минимальная масса звезды самого лёгкого класса равна единице, минимальная масса звезды каждого последующего класса на 1 больше максимальной массы звезды предыдущего класса.

Считается, что таблицу соответствия классов звёзд буквам и границы интервалов хороший астронавигатор должен знать наизусть, так что вопрос про таблицу может попасться на экзамене ордена джедаев.

Люк Скайуокер вытащил тот самый билет... при этом он вообще про существование таблицы узнал впервые. Ни количество классов, ни набор букв, ни их порядок, ни соответствующие им числа ему раньше не встречались. Он знает только верхнюю границу самого тяжёлого класса, а также то, что нижняя граница самого лёгкого класса равна 1.

К счастью, магистр Йода, который принимает экзамен, понимает некоторую избыточность требования дословного знания таблицы и готов ответить на не более, чем 1600 вопросов вида «какой класс имеет звезда заданной массы». Люк хочет подобрать вопросы так, чтобы после того, как получит ответы на них, он смог полностью восстановить таблицу. Помогите ему сделать это.

Протокол взаимодействия

Взаимодействие начинает программа жюри, выводя одно целое число $maxw$ ($2 \leqslant maxw \leqslant 10^{18}$ — верхнюю границу массы звёзд самого тяжёлого класса.

Затем ваша программа задаёт запросы. Каждый запрос имеет вид `?` x, где $1 \leqslant x \leqslant maxw$ — вес звезды, класс которой Люк хочет узнать. В ответ на запрос программа жюри выдаёт одну строчную латинскую букву, задающую соответствующий класс.

Если Вы готовы вывести ответ, выводите `!`, после чего выводите таблицу, отсортированную по возрастанию. В таблице сначала идёт буква, обозначающая класс звезды, затем максимальная масса звезды для этого класса. Все значения в таблице разделяются единичными пробелами. Вывод ответа запросом не считается.

Гарантируется, что количество классов не превосходит 26 и что таблица не меняется в течение всего взаимодействия (то есть что интерактор не адаптивный).

Пример

стандартный ввод	стандартный вывод
3 t s m	 ? 2 ? 3 ? 1 ! m 1 t 2 s 3

Замечание

Не забудьте после каждого запроса или после вывода ответа выводить символ перевода строки, а также сбрасывать буфер ввода-вывода с помощью вызова функции `flush` используемого вами языка программирования.

Задача L. Красим и дополняем

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

У вас есть корневое дерево из n вершин. Корень дерева находится в вершине 1. Вы его хотите покрасить в k цветов. Сначала вы добавляете новые ребра следующим образом – новое ребро между двумя вершинами добавляется тогда и только тогда, когда у этих вершин есть общий родитель. Вы красите получившийся граф так, чтобы соседние вершины имели разный цвет. Стоимость покраски любой вершины в цвет i равна c_i. Вам дано также дополнительное условие: в массиве c присутствует ровно два различных значения. То есть, существуют такие числа $x \neq y$, что $c_i = x$ или $c_i = y$ для любого i. Ваша задача – найти минимальную стоимость корректной раскраски.

Формат входных данных

В первой строке вам дано число $1 \leqslant t \leqslant 10^5$ – количество тестовых примеров.

Далее идет описание самих тестовых примеров. Каждый тестовый пример начинается с чисел $2 \leqslant n \leqslant 2 \cdot 10^5$ – количества вершин в дереве и $2 \leqslant k \leqslant n$ – количества цветов. Гарантируется, что значение k достаточно для того, чтобы хотя бы одна корректная раскраска существовала. В следующих $n-1$ строках задано описание дерева – в каждой строке заданы два числа $1 \leqslant u, v \leqslant n$, $u \neq v$. Гарантируется, что все ребра задают дерево. В следующей строке заданы k натуральных чисел $1 \leqslant c_i \leqslant 10^9$ – стоимости покраски. Гарантируется, что сумма n по всем тестовым наборам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите одно число – минимальную стоимость раскраски графа.

Пример

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

Задача M. Леопольд и множества

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

У вас есть n множеств $p_1, p_2, ..., p_n$, состоящих из натуральных чисел от 1 до n.

Леопольд играет с ними в следующую игру. Сначала он выбирает некоторое множество чисел s от 1 до n(это множество состоит из различных чисел и необязательно содержится в наборе). По нему он хочет составить новое мультимножество q(в этом множестве числа уже могут повторяться). Чтобы сделать это, Леопольд для каждого $x \in s$ рассматривает множество p_x, выбирает некоторый $y \in p_x$ и добавляет y в q. Таким образом, он получает новое мультимножество q, после чего работает уже с ним. Леопольд может делать такие ходы бесконечное множество раз. Таким образом, на каждом ходу он из текущего мультимножества получает новое мультимножество, исключения составляет первый ход, в котором он начинает с множества(а не мультимножества).

Леопольд задался вопросом – а сколько способов существует выбрать некоторое множество s в начале так, что существует непустая последовательность операций, после которой он снова в некоторый момент времени получит множество s?

Так как ответ может получиться большим, выведите остаток от его деления на 998 244 353. Два множества считаются различными, если существует элемент, принадлежащий одному из множеств и не принадлежащий другому.

Формат входных данных

В первой строке содержится количество тестовых примеров $1 \leqslant t \leqslant 2 \cdot 10^5$.

Затем следует описание тестовых примеров. Каждый тестовый пример начинается с числа $1 \leqslant n \leqslant 2 \cdot 10^5$. Далее задаются сами множества p_i. i-я из n последующих строк задаёт множество p_i и содержит набор различных целых чисел от 1 до n. Описание набора начинается с числа k – количества чисел в этом наборе, затем – k целых положительных чисел. Гарантируется, что суммарный размер всех множеств по всем тестовым примерам не превосходит $5 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите ответ на задачу по модулю 998 244 353.

Пример

стандартный ввод	стандартный вывод
2 4 2 2 3 2 1 3 1 4 1 4 4 1 2 1 3 1 4 1 4	7 1