

## Задача 1. Билеты

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

Ходислав решил поехать на Всесибирскую олимпиаду поездом из Санкт-Петербурга в Новосибирск. Он рассматривал разные варианты и заметил, что несмотря на отсутствие билетов от начальной до конечной станции на некоторые поезда, на них можно купить несколько билетов и пересаживаться на станциях внутри одного поезда, чтобы всё-таки доехать из Санкт-Петербурга в Новосибирск.

Поездка на поезде из Санкт-Петербурга в Новосибирск состоит из одного или нескольких переездов от станции до станции. Ходиславу известно, какие места в поезде свободны на каждом переезде; помогите ему купить билеты с наименьшим количеством пересадок внутри одного поезда.

### Формат входных данных

В первой строке записаны два целых числа  $P$  и  $S$  — количество переездов и мест в поезде соответственно ( $1 \leq P \leq 1000$ ,  $1 \leq S \leq 10^4$ ).

Следующие  $P$  строк описывают переезды в порядке следования поезда и состоят из  $S$  идущих подряд символов 0 или 1: 0 на  $i$ -й позиции означает, что место с номером  $i$  занято, 1 — свободно.

Гарантируется, что на этом поезде можно доехать из Санкт-Петербурга в Новосибирск, совершая пересадки внутри поезда.

### Формат выходных данных

Требуется вывести единственное целое число — наименьшее количество пересадок внутри одного поезда, которое необходимо сделать Ходиславу для поездки из Санкт-Петербурга в Новосибирск.

### Пример

стандартный ввод	стандартный вывод
2 3 111 010	0
2 3 110 001	1
9 5 00100 11101 10101 10001 00101 01101 01101 01100 01001	2

## Задача 2. Точки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд 10 секунд (для Java)
Ограничение по памяти:	256 мегабайт

Ксюша любит играть в Точки. В эту игру играют на листе бумаги в клетку, клетки которого образованы  $N$  горизонтальными и  $M$  вертикальными прямыми. Пересечения этих прямых называют **пунктами**. В некоторых пунктах нарисована синяя или красная точка.

**Окружением** называется простой многоугольник, вершины которого это синие точки, а каждая сторона многоугольника это сторона или диагональ клетки. **Окружить красные точки** — означает нарисовать окружение так, чтобы эти красные точки оказались внутри окружения.

Однажды, готовясь стать следующим чемпионом мира по Точкам, Ксюша проанализировала интересную позицию. Она отметила некоторые красные точки, которые она хотела бы окружить, но уже нарисованных синих точек может быть для этого не достаточно. Ксюша просит вас нарисовать в минимально возможном количестве пустых пунктов синие точки так, чтобы отмеченные ею красные точки можно было бы окружить.

### Формат входных данных

В первой строке записаны два целых числа  $N$  и  $M$  ( $1 \leq N, M \leq 100$ ).

В следующих  $N$  строках находятся по  $M$  символов разделённых пробелами. Каждый символ кодирует пункт:

- . — пустой пункт
- o — синяя точка
- x — красная точка
- + — отмеченная красная точка, которую хочет окружить Ксюша

### Формат выходных данных

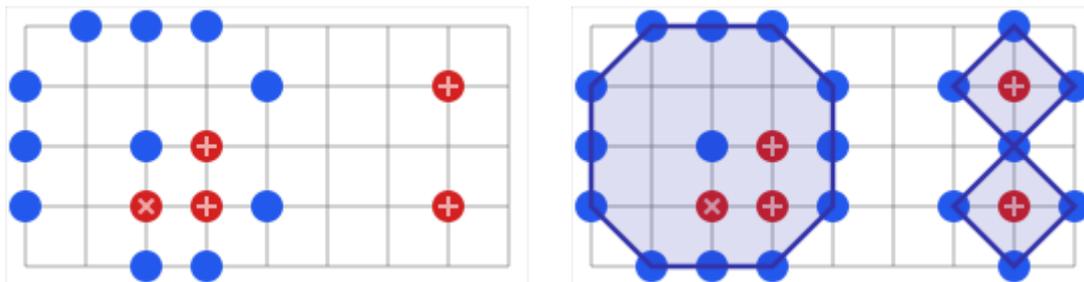
Если все отмеченные точки можно окружить, то выведите получившееся поле с новыми синими точками в том же формате, что и подаётся на вход.

Если окружить все отмеченные точки невозможно, то выведите единственное число  $-1$ .

## Пример

стандартный ввод	стандартный вывод
<pre> 4 5 . . o . x . o + o . . x . . . . . . x .                     </pre>	<pre> 4 5 . . o . x . o + o . . x o . . . . . x .                     </pre>
<pre> 5 9 . o o o . . . . . o . . . o . . + . o . o + . . . . . o . x + o . . + . . . o o . . . . .                     </pre>	<pre> 5 9 . o o o . . . o . o . . . o . o + o o . o + o . . o . o . x + o . o + o . o o o . . . o .                     </pre>
<pre> 3 3 . . . . . + . . .                     </pre>	<pre> -1                     </pre>

Вам рисовать окружения не надо — их нарисует Ксюша. Для второго примера отрисованное условие (слева) и отрисованный ответ вместе с окружениями (справа) может выглядеть вот так



## Задача 3. Кодовый замок

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

Как известно, Ричард Фейнман умел взламывать замки и очень любил это дело. Зная это, его коллега по Манхеттенскому проекту решил применить новый тип замка, надеясь, что на этот раз Фейнману так легко его взломать не удастся.

Новый тип замка содержит матрицу из  $N$  строк и  $M$  столбцов, элементами которой являются нули и единицы. Замок открывается тогда и только тогда, когда сумма чисел в каждой из строк матрицы нечётна. Добиться этого можно следующим преобразованием матрицы: для каждого столбца матрицы выбирается число от 0 до  $N - 1$  и выполняется циклический сдвиг этого столбца на выбранное число. Увидев этот замок, Фейнман сразу понял, что может существовать далеко не единственный способ открыть его. Теперь его занимает подсчёт количества способов, которым можно открыть данный замок.

Давайте, мы тоже посчитаем это количество.

### Формат входных данных

В первой строке записаны два целых числа  $N$  и  $M$  ( $2 \leq N \leq 16$ ,  $2 \leq M \leq 200$ ).

Следующие  $N$  строк состоят из  $M$  идущих подряд символов 0 или 1 — элементов матрицы замка.

### Формат выходных данных

Необходимо вывести одно целое число — количество способов открыть замок. Это число может оказаться слишком большим, поэтому нужно посчитать его по модулю  $10^9 + 9$ .

### Пример

стандартный ввод	стандартный вывод
2 3 111 010	4
2 3 111 000	0
4 3 011 110 000 000	8

### Пояснение к примеру

В первом примере есть такие способы открыть замок:  $(0, 0, 0)$ ,  $(0, 1, 0)$ ,  $(1, 0, 1)$  и  $(1, 1, 1)$ .

## Задача 4. Компромиссные тройки

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

Как-то раз Балаганов и Паниковский поспорили о крышечках. Поклонник языка Basic Паниковский был абсолютно уверен, что  $\wedge$  — это возведение в степень, а сишник Балаганов — что  $\wedge$  — это XOR (побитовая логическая операция «исключающее или»).

Остап Бендер застал спор уже в стадии размахивания руками и возгласов «А ты кто такой?». Узнав, в чём дело, он предложил спорщикам три целых неотрицательных числа  $a$ ,  $b$  и  $c$  таких, что  $a + b > 0$  и  $a \wedge b = c$  как в понимании Паниковского, так и в понимании Балаганова. Конфликт немедленно прекратился.

Наблюдавший за всем этим Козлевич назвал такие тройки компромиссными. Предвидя, что это был не последний спор о крышечках, хозяйственный Козлевич решил выяснить для заданных чисел  $x$  и  $y$ , сколько существует различных компромиссных троек  $a$ ,  $b$ ,  $c$ , таких, что  $x \leq a, b, c \leq y$ . Тройки, отличающиеся перестановкой чисел, считать различными. Например, тройки (29 9 24) и (9 29 24) считаются различными.

### Формат входных данных

Первая строка входных данных содержит одно целое число  $x$  — нижнюю границу диапазона ( $0 \leq x \leq 10^6$ ).

Вторая строка входных данных содержит одно целое число  $y$  — верхнюю границу диапазона ( $x \leq y \leq 10^6$ ).

### Формат выходных данных

Выведите одно целое число — количество компромиссных троек между  $x$  и  $y$  включительно.

### Примеры

стандартный ввод	стандартный вывод
0	1
1	
1	0
1	

## Задача 5. Экзамен по вождению

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

Вы сдаёте экзамен по вождению, состоящий из двух этапов: вождение по городу и ответы на теоретические вопросы. Первым этапом стоит город. Город, где вам предстоит сдать экзамен, представляет собой набор площадей, соединённых двухсторонними дорогами. Известно, что существует способ проехать между любыми двумя площадями, а дорог на одну меньше, чем площадей. Также на каждой площади стоит биллборд, на котором написана ровно одна буква и одно число.

Экзамен проходит по следующим правилам. Вначале инспектор сообщает вам своё имя. Затем вы выбираете начальную и конечную площади и едете по соединяющему их маршруту, не заезжая ни на какую площадь дважды.

В процессе вашего движения инспектор рассчитывает по буквам на биллбордах прочесть своё имя. Если на биллборде, стоящем на очередной площади, которую вы проезжаете, написана очередная буква имени инспектора, то он радуется и не начисляет вам штрафных баллов. Если же там написана любая другая буква, то инспектор злится и прибавляет к вашим штрафным баллам то число, которое написано на этом же биллборде. В любом случае, когда вы проедете площадь, инспектор перейдёт к ожиданию следующей по порядку буквы.

Так как количество штрафных баллов, которое в итоге вы наберёте, будет соответствовать количеству вопросов, на которые вам придётся отвечать во второй части экзамена, вы заинтересованы это число минимизировать. Если вы закончите движение раньше или позже, чем инспектор закончит читать своё имя, то он сочтёт, что вы этот этап экзамена не сдали.

Имя инспектора вам уже известно. Теперь нужно определить оптимальный маршрут.

### Формат входных данных

В первой строке записано целое число  $N$  — количество площадей в городе ( $2 \leq N \leq 2000$ ).

Вторая строка содержит  $N$  строчных латинских букв,  $i$ -ая буква — это та буква, которая написана на биллборде, стоящем на  $i$ -ой площади.

Третья строка содержит  $N$  целых неотрицательных чисел,  $i$ -ое число — это то число, которое написано на биллборде, стоящем на  $i$ -ой площади (все числа не превосходят 1000).

Следующие  $N - 1$  строк описывают дороги в городе: каждая строка содержит два числа — номера площадей, которые эта дорога соединяет (площади нумеруются, начиная с единицы).

В последней строке содержится последовательность из строчных латинских букв — имя инспектора. Длина имени не превосходит 2000 символов.

### Формат выходных данных

В первую строку выведите число, равное минимальному количеству штрафных баллов, которые вы можете набрать на экзамене. В следующую строку выведите номера начальной и конечной площадей в вашем маршруте.

Если вождение по данному городу с этим инспектором сдать невозможно, то выведите единственное число  $-1$ .

## Пример

стандартный ввод	стандартный вывод
7 a b b b a b a 0 2 0 1 0 3 0 1 2 2 3 3 4 4 5 3 6 6 7 aaa	1 5 3
4 a b a b 1 2 3 4 1 2 2 3 3 4 aaaa	6 4 1
4 b a a a 2 1 1 1 1 2 1 3 1 4 aaaa	-1

## Задача 6. База доменов

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

В рамках этой задачи требуется реализовать версиюированную базу, содержащую набор доменов.

Домен — это строка из строчных латинских букв, разделённых точками, точка не может быть первым или последним символом, также в домене не может быть двух точек подряд. Непосредственный поддомен домена  $S$  — это строка  $P.S$ , где  $P$  — строка без точек. Поддомен домена  $S$  — это либо его непосредственный поддомен, либо поддомен его непосредственного поддомена.

База, которую нужно реализовать, должна уметь обрабатывать три типа запросов:

- запрос на добавление домена — создаёт новую версию базы, в которой добавлен указанный в запросе домен;
- запрос на удаление домена — создаёт новую версию базы, в которой указанный в запросе домен удалён;
- запрос на проверку домена — должен дать положительный ответ, если указанный в запросе домен содержится в базе сам или является поддоменом какого-либо содержащегося в базе домена.

### Формат входных данных

В первой строке входного файла содержится целое число  $N$  ( $N \leq 10^5$ ). Далее следуют  $N$  строк с запросами.

Каждая такая строка содержит число  $V$  — версию базы, к которой нужно применить запрос, один из символов «+», «-» или «?» — обозначающий запрос на добавление, удаление или проверку домена соответственно, а также строку — аргумент запроса (домен).

Для любого запроса на добавление гарантируется, что в версии базы, к которой он применяется, добавляемого домена нет; для любого запроса на удаление гарантируется, что в версии базы, к которой он применяется, удаляемый домен есть.

Длина каждого домена не более 1000. Суммарная длина доменов по всем запросам не превосходит  $2 \cdot 10^6$ .

### Формат выходных данных

Для каждого запроса выведите результат его обработки в отдельной строке. Для запросов, модифицирующих базу доменов, выведите число — номер новой версии базы (версии нумеруются последовательно, изначальная версия с номером 0 — это пустой набор доменов).

Для запросов на проверку домена выведите «Y», если данный домен содержится в базе или является поддоменом содержащегося в базе домена и «N» в противном случае.

стандартный ввод	стандартный вывод
25	1
0 + nsu.ru	2
1 + nsuts.ru	3
2 + iis.nsk.su	Y
3 ? fit.nsu.ru	Y
3 ? mmf.nsu.ru	N
3 ? nsk.su	Y
3 ? iis.nsk.su	Y
3 ? www.iis.nsk.su	N
3 ? ya.ru	4
3 - nsu.ru	N
4 ? fit.nsu.ru	N
4 ? mmf.nsu.ru	N
4 ? nsu.ru	Y
4 ? iis.nsk.su	Y
4 ? www.iis.nsk.su	Y
4 ? fresh.nsuts.ru	5
2 + fit.nsu.ru	Y
5 ? nsu.ru	Y
5 ? fit.nsu.ru	Y
5 ? mmf.nsu.ru	N
5 ? iis.nsk.su	6
5 - nsu.ru	N
6 ? nsu.ru	Y
6 ? fit.nsu.ru	N
6 ? mmf.nsu.ru	

## Задача 7. Справедливый делёж

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

Дети лейтенанта Шмидта делят последнюю лакомую территорию — Среднерусскую возвышенность. Причем поделить ее площадь необходимо не поровну, а по справедливости, а именно, в той пропорции, как постановил председатель правления, Шура Балаганов. Если кто не знает, данный географический объект имеет форму круга, резать решили максимально просто, горизонтальными параллельными разрезами.

Ваша задача — подобрать положение этих разрезов.

### Формат входных данных

В первой строке входного файла даны два целых числа — радиус круга  $R$  и количество наследников лейтенанта  $N$  ( $1 \leq R \leq 100, 2 \leq N \leq 100$ ).

В следующей строке указаны пропорции, в которых надо разделить территорию, это  $N$  целых чисел  $x_i$ , записанных через пробел ( $1 \leq x_i \leq 100, 1 \leq i \leq N$ ).

### Формат выходных данных

В выходной файл нужно вывести  $N - 1$  вещественное число, по одному в строке, — высоты, на которых проведены горизонтальные прямые, которые разрезают круг радиуса  $R$  с центром в начале координат в заданных пропорциях. А именно, площади частей должны относиться, как  $x_1, x_2$  и т.д., начиная с верхней.

В ответе допускается погрешность не более, чем  $10^{-5}$  по абсолютному или относительному значению.

### Пример

стандартный ввод	стандартный вывод
5 3	0.000000000
3 2 1	-2.766463562

## Задача 8. Чётные строки

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

Дана строка, состоящая из строчных латинских букв. Требуется заменить все буквы цифрами от 0 до 9 таким образом, чтобы одинаковые буквы были заменены одинаковыми цифрами, разные буквы — разными цифрами, а получившееся число было чётным.

Определите, возможно ли это сделать, и, если возможно, выведите максимально возможное число, которое при этом получится.

### Формат входных данных

Входные данные состоят из одной строки, составленной из строчных латинских букв. Строка непуста и её длина не превосходит  $10^4$ .

### Формат выходных данных

Если заменить буквы цифрами требуемым образом невозможно, выведите  $-1$ . Иначе выведите максимально возможное число, которое получится при замене.

### Примеры

стандартный ввод	стандартный вывод
online	987684
qualifications	-1

## Задача 9. Межгалактическая Лига Чемпионов

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

По заказу отделения Федерации Спортивного Дворового Футбола Васюков Остап Бендер разрабатывает правила межгалактической Лиги Чемпионов по дворовому футболу. За основу он решил взять схему, по которой с текущего сезона играет осенний этап Лиги Чемпионов УЕФА. Схема устроена следующим образом.

- Команды разбиваются по силе на  $k$  непересекающихся корзин по  $n$  команд в каждой (в реальной Лиге Чемпионов УЕФА  $k = 4$ ,  $n = 9$ ).
- Каждая команда играет ровно по два матча с различными командами из каждой корзины (включая и свою): один матч дома и один матч в гостях. То есть команда  $a_i$  из  $i$ -й корзины должна сыграть матч с командой  $b_j$  из  $j$ -й корзины дома и с командой  $c_j$  из  $j$ -й корзины в гостях ( $1 \leq j \leq k$ , ну и разумеется, команды  $a_i$ ,  $b_j$  и  $c_j$  попарно различны); таким образом, каждая команда всего сыграет по  $2k$  матчей.

Кроме того, в схеме от УЕФА есть дополнительные ограничения, связанные с командами из одной лиги, но, так как в проекте Остапа действует правило «одна команда — один двор», все участвующие команды и так представляют разные лиги, так что эти ограничения неактуальны.

По заданным  $n$  и  $k$  выясните, сколько различных вариантов построения турнирной сетки существует. Два варианта считаются различными, если хотя бы у одной команды отличается список команд, с которыми она играет, а также если список совпадает, но в одном варианте команда играет соответствующий матч дома, а в другом — в гостях.

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

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $k$  ( $3 \leq n \leq 10^5$ ,  $1 \leq k \leq 10^5$ ).

### Формат выходных данных

Выведите одно целое число — остаток от деления количества различных вариантов построения турнирной сетки на 998 244 353.

### Примеры

стандартный ввод	стандартный вывод
4 1	6
4 4	501486713

## Задача 10. Круглость

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

Назовём *круглостью* целого положительного числа  $x$ , записанного в системе счисления с основанием  $b > 1$  без ведущих нулей, отношение количества нулей в *конце* записи числа к общему количеству разрядов в его записи.

Для заданного  $x$  найти его максимальную круглость при всех возможных вариантах выбора  $b$ .

### Формат входных данных

Входные данные содержат одно целое число  $x$ , записанное в десятичной системе счисления ( $1 \leq x \leq 10^{12}$ ).

### Формат выходных данных

Выведите два взаимно простых целых числа  $p \geq 0$  и  $q > 0$  такие, что максимальная круглость числа  $x$  при правильном выборе системы счисления равна  $p/q$ .

### Примеры

стандартный ввод	стандартный вывод
1	0 1
4	2 3
24	3 5

## Задача 11. Угадайте многоугольник

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

Это — интерактивная задача

Программа жюри загадала правильный  $n$ -угольник ( $3 \leq n \leq 100$ ), вписанный в окружность целого радиуса  $R$  ( $1 \leq R \leq 100$ ). Вы можете задавать запросы «сравните заданное число  $p/q$ , где  $p$  — целое число, а  $q$  — степень двойки, и отношение площади к периметру».

Программа жюри сравнивает числа с абсолютной погрешностью  $10^{-6}$  и выдаёт результат сравнения (больше, меньше или равно).

Ваша задача — угадать как  $n$ , так и  $R$  не более, чем за 14 запросов.

### Протокол взаимодействия

Взаимодействие начинается программа жюри, которая выводит количество сценариев  $t$  ( $1 \leq t \leq 200$ ). Далее один за другим проходят  $t$  сценариев взаимодействия.

В каждом сценарии взаимодействие начинается ваша программа, задавая вопрос в формате ?  $p$   $q$ , где  $0 \leq p \leq 2^{31} - 1$ ,  $0 \leq q \leq 24$ . Ответом на вопрос будет:

- <, если  $p/2^q$  меньше отношения площади к периметру на  $10^{-6}$  или более,
- >, если  $p/2^q$  больше отношения площади к периметру на  $10^{-6}$  или более,
- = в остальных случаях.

Как только вы готовы вывести ответ, выведите !  $n$   $R$ , где  $R$  — значение радиуса, а  $n$  — количество сторон многоугольника. Вывод ответа в общем количестве запросов не учитывается.

Если ответ правильный, программа жюри выведет gg, после чего ваша программа должна перейти к обработке следующего сценария или завершить выполнение, если закончившийся сценарий — последний.

При неправильном ответе или нарушении протокола выполнение программы будет прервано, и вы получите соответствующий вердикт.

Гарантируется, что параметры  $n$  и  $R$  определяются перед началом каждого сценария и не изменяются в процессе взаимодействия (то есть что интерактор не является адаптивным).

### Пример

стандартный ввод	стандартный вывод
1	? 1 0
<	? 10 1
>	? 10 2
=	! 3 10
gg	

### Замечание

Не забывайте после вывода запроса или ответа выводить перевод строки, а также сбрасывать буфер ввода-вывода вызовом функции `flush` используемого вами языка программирования.

## Задача 12. Одним пальцем

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

Рассмотрим процесс набора одним пальцем длинных чисел на обычной клавиатуре и цифровом блоке. Числа состоят из цифр от 1 до 9 включительно.

При наборе двух соседних цифр палец проходит расстояние, равное евклидову расстоянию между центрами соответствующих клавиш. Клавиши являются квадратами со стороной 1. Расстоянием между клавишами можно пренебречь.

Клавиши в обычной клавиатуре расположены в виде прямоугольника  $1 \times 9$  со следующей раскладкой:

123456789

Клавиши в цифровом блоке расположены в виде прямоугольника  $3 \times 3$  со следующей раскладкой:

789

456

123

Время набора числа одним пальцем зависит от суммарного расстояния, пройденного пальцем — если расстояние меньше, то и время набора меньше. Считается, что первоначально палец сразу установлен на первую цифру и что сами нажатия происходят мгновенно, то есть число, состоящее из одинаковых цифр, набирается за время 0.

Вам дано длинное число, требуется выяснить, на чём его быстрее набирать — только на обычной клавиатуре или только на цифровой.

### Формат входных данных

Входные данные содержат одну строку, состоящую из не менее 2 и не более  $10^7$  цифр от 1 до 9 включительно — число, которое требуется ввести.

### Формат выходных данных

Выведите `num`, если число быстрее набрать на цифровом блоке, `usual`, если на обычной клавиатуре, или `same`, если там и там число набирается за одинаковое время.

### Примеры

стандартный ввод	стандартный вывод
144	num
121	same
343	usual

## Задача 13. Римская запись, двоичная запись...

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

*Это интерактивная задача*

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

Заметим, что:

- 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).

Загадано число от 1 до 3999 включительно. Вы можете задавать следующие запросы:

- `s` — спросить сумму римских цифр этого числа (то есть сумму значений всех однобуквенных подстрок его римской записи).
- `? i` — спросить значение  $i$ -го бита в двоичной записи этого числа ( $0 \leq i \leq 11$ ).

Ваша задача — отгадать число не более, чем за четыре операции.

### Протокол взаимодействия

Взаимодействие начинает программа жюри, выводя одно целое число  $t$  — количество сценариев ( $1 \leq t \leq 100$ ).

В каждом сценарии ваша программа задаёт запросы в формате `s` — для запроса суммы римских цифр (в ответ выдаётся одно число, записанное в обычной десятичной системе счисления), `? i` ( $0 \leq i \leq 11$ ) — для запроса, чему равен  $i$ -й самый младший бит в двоичной записи загаданного числа (ответ 0 или 1). Всего можно задать не более четырёх запросов за сценарий.

Как только ваша программа может точно назвать загаданное число, она выводит `! s`, где `s` — запись загаданного числа в виде римского числа. Если число угадано, то программа жюри выводит сообщение `gg`, после чего ваша программа должна перейти к следующему сценарию (или завершить выполнение, если сценарий последний). Вывод ответа в общем количестве запросов не учитывается.

При неправильном ответе или нарушении протокола выполнение программы будет прервано, и вы получите соответствующий вердикт.

Гарантируется, что загаданное число определяется перед началом сценария и не изменяется в процессе игры (то есть что интерактор не адаптивный).

## Пример

стандартный ввод	стандартный вывод
1	? 0
0	? 7
1	s
600	! CD
gg	

## Замечание

Не забывайте после вывода запроса или ответа выводить перевод строки, а также сбрасывать буфер ввода-вывода вызовом функции `flush` используемого вами языка программирования.