Задача A. Угадай число

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

Это интерактивная задача. В процессе тестирования программа участника взаимодействует с программой жюри с использованием стандартных потоков ввода/вывода.

Программа жюри загадала число от 1 до n, цель программы участника — отгадать его, задав не более чем 30 вопросов. Для этого программа участника сообщает свои догадки программе жюри, а программа жюри отвечает, является ли загаданное число бóльшим, меньшим или равным сделанной догадке.

Протокол взаимодействия

Сначала необходимо прочитать из стандартного потока ввода число n ($1 \leqslant n \leqslant 10^9$). Затем протокол общения следующий: требуется вывести в стандартный поток вывода одну строку, содержащую целое число от 1 до n — свою догадку о загаданном числе.

После этого необходимо считать из стандартного потока ввода одно число: сообщение программы жюри. Возможны следующие сообщения:

- «`1`» — загаданное число больше последней догадки;
- «`-1`» — загаданное число меньше последней догадки;
- «`0`» — последняя догадка верна. Считав 0, ваша программа должна завершиться.

Обратите внимание на необходимость перевода строки после каждой выведенной догадки, прочитайте подробности про интерактивные задачи в памятке участника.

Пример

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

В примере ввод и вывод отформатированы пустыми строками, чтобы было видно, какие запросы соответствуют каким ответам программы жюри. В реальном взаимодействии необходимо переводить строку после каждого запроса, но выводить пустые строки не надо.

Задача B. Одинокое число

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

Это интерактивная задача.

Однажды Маше было нечего делать, и она записала на листе бумаги N целых чисел: $a_1, a_2, \ldots, a_N$. Маша недавно изучила алгоритмы сортировки, поэтому она выписала свои числа в неубывающем порядке, то есть $a_1 \leqslant a_2 \leqslant \ldots \leqslant a_N$.

Также Маша очень любит загадки, поэтому среди ее чисел есть некоторое число C, которое встречается среди выписанных чисел ровно один раз, а все остальные числа встречаются ровно два раза.

Маша загадала вам загадку — найти «одинокое» число C. Для этого вы можете не более, чем 42 раза попросить Машу сообщить вам i-е записанное число.

Маша сообщила вам, что $1 \leqslant a_i \leqslant 10^9$.

Протокол взаимодействия

В начале ваше решение должно считать число N ($1 \leqslant N \leqslant 10^6$) — количество записанных Машей чисел.

Затем ваше решение может сделать не более 42 запроса. Для того, чтобы сделать запрос, ваше решение должно вывести его в следующем формате: «`?` i» (без кавычек, $1 \leqslant i \leqslant N$). Ответом на запрос является число a_i.

Для того, чтобы вывести ответ, ваше решение должно вывести «`!` C», после чего немедленно завершить работу.

Вы должны в точности соблюдать протокол взаимодействия с интерактором, в противном случае решение может получить произвольный вердикт.

При превышении числа запросов вы получите вердикт «`Неправильный ответ`».

После каждого запроса, в том числе после вывода ответа, вы должны выполнить операцию `flush`.

Для сброса буфера вывода (то есть для операции «`flush`») сразу после вывода запроса и перевода строки нужно сделать:

- fflush(stdout) или cout.flush() в языке C++;
- System.out.flush() в Java;
- sys.stdout.flush() в Python;
- flush(output) в Pascal;
- смотрите документацию для других языков.

Если вы не сделаете операцию `flush` после какого-либо запроса, ваше решение может получить любой вердикт.

Пример

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

Замечание

В примере записанные числа равны: $1, 1, 4, 5, 5$. Одиноким числом, конечно, является число 4.

Задача C. В поисках истины

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

Это интерактивная задача.

После вывода очередной строки не забывайте очищать буфер потока вывода после каждого запроса. Для этого можно, например, воспользоваться командами fflush(stdout), cout.flush(), либо выполнять перевод строки при помощи endl в C++, system.out.flush() в Java, flush(output) в Pascal или sys.stdout.flush() в Python.

Сарком управлял совет Великих сквайров. В совете состояло пятеро саркитов, и самый богатый и влиятельный из них, сквайр Файф, взял на себя ответственность разобраться в истории космоаналитика. Волей случая к нему попал и Рик, который оказался космоаналитиком, а Селим Джунц и Людиган Эбл из посольства Трантора были готовы с ним сотрудничать. Сквайра интересовало одно — кто же мог совершить столь ужасное преступление?

Еще раньше Великому сквайру поступали анонимные письма, в которых сообщалось, что Флорина должна погибнуть. Шантажист требовал отдать значительную долю кыртовых полей Файфа за эту информацию. Сквайр подозревал, что если преступник смог похитить космоаналитика, прятать его целый год от властей и при этом шантажировать самого главного человека на всем Сарке, он тоже должен быть Великим сквайром. И больше всего подозрений вызывал у него сквайр Стин.

Тут неожиданно Рик вспомнил, что во время разговора с похитителем, тот сообщил размер его владений на Флорине — k квадратных километров. В архиве хранятся данные о площади земель s_i, контролируемых сквайром Стином, в n моментов времени. Файфу известно, что все s_i различны, а так же что до некоторого момента эти площади увеличивались, а потом начали убывать. Более формально, существует такое $1 \leqslant t \leqslant n$, что для любого $1 < i \leqslant t$ $s_{i-1} < s_i$ и для любого $t < j \leqslant n$ $s_{j-1} > s_j$. Чтобы подтвердить свою правоту, Великому сквайру нужно узнать, в какой момент времени площадь владений Стина в точности равнялась k, и он просит вас помочь ему. Вы можете по моменту времени i узнать размер владений сквайра Стина s_i. Небольшая сложность состоит в том, что времени у Файфа мало, а направление запроса в архив происходит достаточно долго. Поэтому у вас есть возможность задать не более 80 таких запросов. Сквайр не сомневается в своем в успехе, и гарантирует вам, что искомое s_i существует.

Протокол взаимодействия

Изначально вам заданы два числа n, k — количество записей о владениях Стина в архиве и значение площади, которое интересует Файфа ($2 \leqslant n \leqslant 2 \cdot 10^5, 0 \leqslant k \leqslant 10^9$). Далее ваша программа может делать запросы вида “? i”, в качестве ответа на которые программа жюри будет выводить значения s_i. Все s_i - целые числа, $0 \leqslant s_i \leqslant 10^9$. Записи в архиве нумеруются с 1. Когда ваша программа будет готова дать ответ, она должна вывести “! i”, где $s_i = k$, и завершиться. Если ваша программа сделает больше 80 запросов первого типа, решение получит вердикт “Wrong answer”. Если программа не завершится после запроса второго типа или ответ на запрос второго типа будет неверным, решение также получит вердикт “Wrong answer”.

Пример

стандартный ввод	стандартный вывод
5 3 2 8 10 1 3	? 5 ? 4 ? 3 ? 1 ? 2 ! 2

Замечание

В данном примере записи о владениях выглядели так: $1, 3, 10, 8, 2$.

Задача D. Максимальная медиана

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

У вас есть массив a длины n. Найдите подотрезок $a[l \dots r]$ длиной хотя бы k, имеющий максимально возможную медиану.

Медианой нечётного по размеру массива называется средний элемент, если массив отсортировать по неубыванию. Медианой четного по размеру массива называется меньший из двух средних элементов.

Формат входных данных

В первой строке находятся два целых числа n и k ($1 \leqslant k \leqslant n \leqslant 10^6$).

Во второй строке находятся n целых чисел a_1, a_2, $\dots$, a_n ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите одно целое число m — максимальную возможную медиану.

Примеры

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

Задача E. Доставщик пиццы

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

В государстве находится n офисов, в каждом из которых работает множество сотрудников. И сотрудники каждого офиса могут загрустить! Доставщики пиццы об этом знают, и они не хотят потерять возможность поднять настроение офисным сотрудникам, продав им много больших пицц.

Сейчас доставщики пицц планируют построить самую большую пиццерию в государстве, однако они еще не определились с ее местоположением. Поскольку для доставщиков пицц каждый километр дается значительно тяжелее предыдущего, доставщики хотят найти такую точку в государстве, из которой можно было бы добраться до самого удаленного офиса за наименьшее возможное время.

Государство не обычное, а четырехмерное. Поверхностью планеты, на которой находится государство, является трехмерное пространство. Разумеется, все офисы расположены на данном трехмерном пространстве.

Расстояние между точками $(x_i,\ y_i,\ z_i)$ и $(x_j,\ y_j,\ z_j)$ вычисляется по формуле $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$. Пиццерию можно строить в любой точке, даже в каком-либо из офисов.

Экономика государства в опасности! Сообщите, где необходимо построить пиццерию, чтобы всем жилось хорошо!

Формат входных данных

В первой строке входного файла содержится целое число n — количество офисов ($1 \leqslant n \leqslant 100$). Каждая из последующих n строк содержит информация о офисах. i-я из этих строк содержит три целых числа x_i, y_i, z_i — координаты i-го офиса ($-10^4 \leqslant x_i,\ y_i\ z_i \leqslant 10^4$). Никакие два офиса не совпадают.

Формат выходных данных

В первой строке выходного файла следует вывести три вещественных числа через пробел x_0, y_0, z_0 — координаты будущей пиццерии. Если существует несколько решений, то разрешается вывести любое. Ответ будет засчитан, если расстояние от данной точки до самого удаленного офиса будет отличаться от результата жюри не более чем на 10^{-6} по абсолютному или относительному значению.

Пример

стандартный ввод	стандартный вывод
5 4 -6 -8 3 -10 -3 3 5 2 -2 3 10 -1 -9 -8	-1.2621406 -2.8964323 0.9441693

Замечание

В примере максимальное расстояние от базы до какой-либо из планет примерно равняется 10.8314551386.

Задача F. Общее число

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

Введем определение функции $f(x)$:

$$f(x) = \begin{cases} \frac{x}{2} \text{ если } x \text{ четное} \\ x - 1 \text{ в противном случае} \end{cases}$$

Можно заметить, что если мы выбираем некоторое число v и применяем к нему функцию f, затем применяем функцию f к числу $f(v)$, и так далее, то рано или поздно мы получим значение 1. Выпишем все промежуточные значения на пути от v до 1 и назовем все выписанные значения $path(v)$. Например $path(1) = [1]$, $path(15) = [15, 14, 7, 6, 3, 2, 1]$, $path(32) = [32, 16, 8, 4, 2, 1]$.

Выпишем все $path(x)$ для всех x от 1 до n. Перед вами стоит задача: определить максимальное число m такое, что m встречается хотя бы в k различных $path(x)$.

Иначе говоря, нужно найти такое максимальное y, что

$$|\{x \mid 1 \leqslant x \leqslant n,\ y \in path(x)\}| \geqslant k$$

.

Формат входных данных

В первой строке следуют два целых числа n и k ($1 \leqslant k \leqslant n \leqslant 10^{18}$).

Формат выходных данных

Выведите максимальное число m такое, что m встречается хотя бы в k различных $path(x)$.

Примеры

стандартный ввод	стандартный вывод
11 3	5
11 6	4
20 20	1
14 5	6
1000000 100	31248

Задача G. Угадай массив

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача. Ваша программа будет взаимодействовать с программой жюри, используя стандартный ввод и вывод.

Алиса и Боб решили сыграть в игру под названием «Угадай массив». Правила игры очень простые: Алиса загадала массив целых чисел размера n, а Бобу нужно угадать этот массив, сделав не больше n запросов о сумме на отрезке.

За один ход Боб может сделать к Алисе запрос одного из двух типов:

1. «`? l r`» — узнать сумму чисел на отрезке массива с l-го по r-й элемент включительно;
2. «`!`» — сообщить Алисе, что он готов дать ответ. После этого запроса Алиса ожидает от Боба n целых чисел — загаданный массив.

На каждый запрос первого типа, Алиса говорит Бобу сумму чисел на запрошенном отрезке. Но чтобы отгадывать было сложнее, после каждого запроса Алиса делает один отрезок *запрещенным*. В дальнейших запросах Боб не может запрашивать сумму чисел на запрещенных отрезках.

Помогите Бобу угадать загаданный массив, сделав не более n запросов первого типа.

Протокол взаимодействия

В первой строке ввода дано целое число n — размер загаданного массива ($1 \leqslant n \leqslant 10^4$). Гарантируется, что все числа в массиве по модулю не превосходят 10^9.

Далее запускается протокол взаимодействия с программой жюри — интерактором.

Интерактор ожидает от вашей программы запросы двух типов: «`? l r`» или «`!`», где l, r — целые числа — границы отрезка, на котором вы хотите узнать сумму ($1 \leqslant l \leqslant r \leqslant n$). После каждого запроса должен следовать перевод строки. При несоблюдении вашей программой формата запросов, ваше решение может получить произвольный вердикт (отличный от `OK`).

После запроса первого типа необходимо считать со стандартного ввода три целых числа s, l_b и r_b — сумму чисел на запрошенном отрезке s и границы нового *запрещенного* отрезка $[l_b, r_b]$, на котором запрашивать сумму больше нельзя ($|s| \leqslant 10^{18}$; $1 \leqslant l_b \leqslant r_b \leqslant n$).

Запрос второго типа означает, что ваша программа готова дать ответ на задачу. После запроса второго типа необходимо вывести n целых чисел — загаданный массив.

Обратите внимание, ваша программа может сделать не более n запросов первого типа. При превышении данного ограничения, а также при попытке узнать сумму на *запрещенном* ранее отрезке, интерактор выведет «`-1 -1 -1`» и завершится с вердиктом `WA`. Чтобы не получить вердикт `TL` или `IL`, считав из ввода значения «`-1 -1 -1`», ваша программа должна завершить свою работу с нулевым кодом возврата.

Пример

стандартный ввод	стандартный вывод
3 1 2 2 3 2 3 6 1 2	? 1 1 ? 3 3 ? 1 3 ! 1 2 3

Задача N. Поиграем?

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это **интерактивная** задача.

1804 год. Вице-президент Соединённых Штатов Аарон Бёрр вызывает на дуэль кандидата в губернаторы штата Нью-Йорк Александра Гамильтона за серию оскорбительных памфлетов в свой адрес.

Но Бёрр разумный человек и понимает, что даже если он убьет Гамильтона на дуэли, он потеряет свою репутацию, и его политическая карьера будет закончена. Поэтому противники решили просто сыграть в игру. Для честности они решили сыграть в неё g раз.

Каждую игру Гамильтон загадывает целое положительное число n, а Бёрр пытается его отгадать. Для любого целого положительного x Бёрр может спросить у Гамильтона долю чисел между 1 и n включительно, которые делятся на x. Иными словами, спрашивая про x, он получает значение выражения

$$\frac{\left\lfloor \frac{n}{x} \right\rfloor}{n},$$

причём Гамильтон сообщает ему результат в виде **несократимой** дроби (здесь $\lfloor r \rfloor$ обозначает результат округления вниз вещественного числа r).

Помогите Бёрру найти ответ за некоторое заранее определённое число запросов.

Протокол взаимодействия

При запуске решения на вход подается целое число g — число игр между Гамильтоном и Бёрром ($1 \leqslant g \leqslant 1000$).

Для каждого теста зафиксировано число q ($6 \leqslant q \leqslant 60$) — максимальное количество запросов в одной игре. Гарантируется, что q запросов достаточно, чтобы решить задачу. Эти числа не сообщаются программе участника, но ограничения на эти числа в различных подзадачах приведены в таблице системы оценивания. Если программа участника делает более q запросов программе жюри, на этом тесте она получает в качестве результата тестирования «Wrong answer».

Чтобы сделать запрос, следует вывести строку «`X` t», где t — целое положительное число ($1 \leqslant t \leqslant 10^{18}$), для которого требуется узнать значение выражения

$$\frac{\left\lfloor \frac{n}{t} \right\rfloor}{n}$$

В ответ на каждый запрос программа получает через пробел два числа a и b — числитель и знаменатель этой дроби **после сокращения** — или число -1 в случае, если программа превысила ограничение по числу запросов.

Если программа определила загаданное число, то она должна вывести строку «`N` t», где t — предполагаемый ответ ($1 \leqslant t \leqslant 10^{18}$). Если ответ был правильный, то в ответ программа получает строку «`Correct`», а если неправильный, то она получает строку «`Wrong`».

После этого программа переходит к следующей игре, если они остались, иначе она должна завершиться.

Обратите внимание, в случае считывания числа -1 или строки «`Wrong`» вы **обязательно** должны сразу завершить вашу программу. В противном случае вердикт тестирующей системы может быть некорректным, в частности, вы можете получить вердикт Run-time error или Time limit exceeded!

Гарантируется, что в каждом тесте загаданные числа фиксированы в самом начале игры и не изменяются в зависимости от ваших запросов.

Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Баллы	Дополнительные ограничения
0	0	$q = 60$
1	30	$q = 60$
2	10	$q = 30$
3	4	$q = 20$
4	4	$q = 15$
5	5	$q = 10$
6	5	$q = 9$
7	11	$q = 8$
8	10	$q = 7$
9	21	$q = 6$

Пример

стандартный ввод	стандартный вывод
2 1 2 3 10 1 5 1 5 1 10 1 10 0 1 Correct 1 1 0 1 Correct	 X 2 X 3 X 5 X 4 X 6 X 10 X 11 N 10 X 1 X 2 N 1

Замечание

В первом примере $g = 2$. Приведены примеры запросов, по которым игрок угадывает, что в первой игре загадано число 10, а во второй 1. Эти же числа загаданы в первом тесте в тестирующей системе.

В точности соблюдайте формат выходных данных. После каждого вывода обязательно выводите один перевод строки и сбрасывайте буфер вывода — для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Задача I. Инверсия и Разворот

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

Это интерактивная задача.

Вы играете со своим другом в крайне увлекательную игру. Ваш друг, будучи программистом, загадал строку, состоящую из символов «0» и «1» длины B.

Вы хотите отгадать эту строку. Для этого вы можете назвать число P ($1 \leqslant P \leqslant B$), после чего ваш друг сообщит P-й символ загаданной строки.

Спустя несколько часов не очень увлекательной игры ваш друг предложил усложнить игру. Пусть вы хотите спросить у вашего друга очередной символ загаданной строки, причем это уже Q-й ваш вопрос. Если число Q оканчивается на цифру «1», то есть равно $1, 11, 21, 31, 41, \ldots$, то ваш друг, прежде чем ответить на ваш вопрос, случайно равновероятно выбирает число от 1 до 4 и выполняет соответствующую операцию:

1. Инвертировать все биты в строке, то есть заменить все символы «0» на «1» и наоборот.
2. Развернуть строку, то есть поменять местами первый и последний символы, второй и предпоследний символы, и так далее.
3. Инвертировать, а затем развернуть строку.
4. Ничего не делать.

После того, как друг выполнил одну из операций, он сообщает вам P-й символ получившейся строки.

Теперь, так как правила игры изменились, вы хотите научиться выигрывать, а именно: используя не более 150 вопросов отгадать строку.

Обратите внимание, вы должны отгадать не изначальную строку, а строку после всех сделанных изменений.

Протокол взаимодействия

Для начала считайте два числа T и B ($1 \leqslant T \leqslant 100$) — количество тестовых случаев и длину загаданной строки.

После этого для каждого тестового случая повторяется следующая процедура. Во всех тестовых случаях длина строки одинаковая, однако сами загаданные строки могут различаться.

Вы можете задавать программе жюри вопросы следующего вида: `?` P. После того, как вы задали вопрос, происходят следующие действия: если ваш вопрос является 1-м, 11-м, 21-м, и так далее, то сначала программа жюри выполняет одну из четырех описанных операций. Номер операции выбирается случайно равновероятно. После этого программа жюри выдаст в качестве ответа символ «0» или «1» (без кавычек).

После того, как вы считаете, что готовы сделать ответ, выведите ответ в формате: `!` S, где S — строка длины B, состоящая из символов «0» и «1». После того, как вы вывели ответ, программа жюри выдаст в качестве ответа символ «Y», если ваш ответ является правильным, либо «N» в противном случае (символы выдаются без кавычек). Если ваш ответ не является правильным, вы должны немедленно завершить программу.

Количество вопросов для каждого тестового случая не должно превосходить 150. В случае несоблюдения описанного протокола взаимодействия с программой жюри, вы можете получить произвольный вердикт.

Обратите внимание, что каждое выведенное вами сообщение должно завершаться переводом строки. Также после вывода каждого сообщения ваша программа должна очищать потоковый буфер, чтобы выведенная вами информация дошла до программы жюри: например, это делают вызовы `fflush(stdout)` или `cout.flush()` в C++, `System.out.flush()` в Java, `Console.Out.Flush()` в C#, `flush(output)` в Pascal, `sys.stdout.flush()` в Python.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	подзадача	–
1	10	$B = 10$	подзадача	–
2	40	$B = 20$	подзадача	–
3	50	$B = 100$	подзадача	1

Пример

стандартный ввод	стандартный вывод
1 12 0 0 0 1 Y	 ? 1 ? 2 ? 3 ? 12 ! 000101110101

Замечание

Пример лишь иллюстрирует протокол взаимодействия. Не гарантируется, что ответ на него действительно такой. Переводы строк в примере выполнены для удобства.

Для начала узнаем первый символ загаданной строки. Так как это первый вопрос, его номер заканчивается на цифру «1», поэтому со строкой произойдет одна из четырех операций, после чего программа жюри сообщает, что первый символ равен 0. Далее мы узнаем второй, третий и последний символ. В конце мы вышли в астрал, узнали правильный ответ и сообщили его программе жюри, на что получили положительный ответ. При решении задачи не пытайтесь выйти в астрал, доверьте это профессионалам.

Задача J. Новогодний и прямоугольный

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

Это **интерактивная** задача.

На Новый год Дед Мороз подарил Глебу то, о чём он уже давно мечтал — клетчатый квадрат размером $n \times n$. Подарок этот не простой, а с сюрпризом — внутри квадрата Дед Мороз выбрал некоторый непустой прямоугольник, и в каждую клетку этого прямоугольника он положил по мандарину.

Теперь, чтобы получить желанный подарок, Глебу нужно сыграть с Дедом Морозом в очень интересную игру. Глеб должен отгадать, в каком именно прямоугольнике находятся все мандаринки, подаренные Дедом Морозом. Будем считать, что строки и столбцы занумерованы числами от 1 до n снизу вверх и слева направо. Глеб может производить два типа запросов:

- ? $x_1\, y_1\, x_2\, y_2$ ($1 \leqslant x_1 \leqslant x_2 \leqslant n$, $1 \leqslant y_1 \leqslant y_2 \leqslant n$) — в ответ на этот запрос Дед Мороз говорит, сколько мандаринок находится в прямоугольнике, левым нижним углом которого является клетка (x_1, y_1), а правым верхним — клетка (x_2, y_2);
- ! $x_1\, y_1\, x_2\, y_2$ ($1 \leqslant x_1 \leqslant x_2 \leqslant n$, $1 \leqslant y_1 \leqslant y_2 \leqslant n$) — когда Глеб уверен, что он точно знает, где находятся мандаринки, он должен сделать запрос такого вида, чтобы сообщить свой ответ. При этом (x_1, y_1) соответствует предполагаемому расположению левого нижнего угла, а (x_2, y_2) — правого верхнего.

Формат входных данных

При запуске решения на вход вашей программе подается одно число n ($1 \leqslant n \leqslant 2 \cdot 10^9$) — размер квадрата.

Затем на каждый запрос типа "?" вам будет выдаваться количество мандаринок, находящихся в указанном вами прямоугольнике.

Формат выходных данных

Вы должны выводить корректные запросы в формате, описанном выше. Последним должен следовать единственный запрос вида "`!`", после чего ваша программа должна немедленно завершиться. Ваша программа должна произвести не больше q (параметр зависит от номера группы) запросов типа "?". Обратите внимание, что последний запрос, выводящий ответ, не входит в данные q запросов.

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода — для этого используйте команды `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Примеры

стандартный ввод	стандартный вывод
4 6 6 4 	? 1 1 4 4 ? 1 3 4 4 ? 2 3 4 4 ! 1 3 3 4

Замечание

Пример в условии иллюстрирует взаимодействие с проверяющей программой. Для прохождения первого теста не обязательно производить такие же запросы, как в примере.

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			n	q	
0	1 – 1	0	$n = 4$	$q = 1000$	Тест из условия.
1	2 – 12	10	$n \leqslant 10$	$q = 10\,000$	
2	13 – 23	20	$n \leqslant 100$	$q = 10\,000$	
3	24 – 34	20	$n \leqslant 10\,000$	$q = 20\,000$	
4	35 – 45	25	$n \leqslant 2 \cdot 10^9$	$q = 128$	
5	46 – ∞	25	$n \leqslant 2 \cdot 10^9$	$q = 64$	

Задача K. Угадай выражение

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

Это интерактивная задача.

Жюри загадало некоторое арифметическое выражение, содержащее $n + 1$ переменных $a_0, a_1, \ldots, a_n$, которое вычисляется по формуле:

$$(\ldots(((a_0\ op_1\ a_1)\ op_2\ a_2)\ op_3\ a_3)\ldots\ op_n\ a_n) \bmod 10^9 + 7,$$

где $op_1, op_2, \ldots, op_n$ — это арифметические операции, каждая из которых является либо операцией «+» (сложение), либо операцией «×» (умножение), а $a_0, a_1, \ldots, a_n$ — переменные.

Например, если значения переменных равны $(a_0, a_1, a_2) = (1, 1, 2)$, а операции равны $(op_1, op_2) = (+, \times)$, то в результате вычисления выражения мы получим значение $((1 + 1) \times 2) \bmod 10^9 + 7 = 4$.

Вы можете несколько раз задать некоторые значения каждой из переменных и попросить жюри вычислить значение выражения. После этого вы должны угадать, чему равны $op_1, op_2, \ldots, op_n$.

Протокол взаимодействия

В начале взаимодействия с программой жюри вы должны считать целое число n ($1 \leqslant n \leqslant 4\,000$) — количество операторов в формуле.

После этого вы можете вывести не более 275 запросов вида: «? $a_0\ a_1\ \ldots\ a_n$» ($0 \leqslant a_i < 10^9 + 7$). После этого вы должны считать одно число — значение выражения для данных переменных.

Когда вы определите, чему равны $op_1, op_2, \ldots, op_n$, вы должны вывести запрос вида: «! s», где s — строка, состоящая из n символов `+` или `x` (маленькая латинская буква x), где i-й символ строки соответствует операции op_i.

Не забывайте выполнять операцию `flush` после каждого запроса (в том числе и после последнего).

В случае, если вы выведите больше 275 запросов на вычисление выражения, либо не будете соблюдать описанный протокол взаимодействия с программой жюри, вы можете получить любой вердикт.

Примеры

стандартный ввод	стандартный вывод
2 4 6	? 1 1 2 ? 1 1 3 ! +x
10 5 6224 640750	? 1 1 1 1 1 1 1 1 1 1 1 ? 0 4 2 4 2 4 2 4 2 4 2 ? 1 2 3 4 5 6 7 8 9 10 11 ! ++xxx+x+xx

Задача L. Игра с бинарной строкой

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

Это интерактивная задача.

Алиса и Боб играют в игру. У них есть строка длины n, каждый символ строки — это либо `0`, либо `1`. Игроки ходят по очереди, начинает Алиса. На каждом ходу игрок должен изменить **ровно** один символ строки: `0` меняется на `1`, а `1` — на `0`. После хода игрока должна получиться строка, которая раньше в игре никогда не встречалась (в том числе до всех ходов). Если игрок не может сделать ход, то он проигрывает.

Вам нужно выбрать, за кого (Алису или Боба) вы хотите играть, проверяющая система будет играть за другого игрока. Вы должны выиграть игру за выбранного игрока.

Протокол взаимодействия

В начале вы должны считать n ($1 \leqslant n \leqslant 15$) — длину строки — и строку s длины n — начальное состояние строки. После этого выведите «`Alice`» (если вы хотите играть за Алису) или «`Bob`» (если хотите играть за Боба).

В каждый свой ход вы должны вывести одно число p ($0 \leqslant p \leqslant n$).

- $p = 0$ символизирует, что вы сдаётесь. $1 \leqslant p \leqslant n$ — позиция символа, который вы меняет своим ходом. Позиции в строке нумеруются слева направо от 1 до n.

В каждый ход соперника вы должны считать одно число p ($-1 \leqslant p \leqslant n$).

- $p = 0$ символизирует, что соперник сдаётся. В этом случае вы должны завершить выполнение программы.
- $p = -1$ символизирует, что ваш последний ход привёл к строке, которая ранее встречалась, либо вы совершили недопустимый ход. В этом случае вы также должны завершить выполнение программы, иначе вы можете получить произвольный вердикт. $1 \leqslant p \leqslant n$ — позиция символа, который соперник меняет своим ходом.

Обратите внимание, что если вы выбрали Алису, то вы делаете первый ход, а если вы выбрали Боба, то вы делаете второй ход.

Не забудьте после каждого хода выполнять операцию '`flush`'.

Для сброса буфера вывода (то есть для операции '`flush`') сразу после вывода хода и перевода строки нужно сделать:

- fflush(stdout) в языке C++;
- System.out.flush() в Java;
- stdout.flush() в Python;
- flush(output) в Pascal;
- смотрите документацию для других языков.

В случае, если вы не будете выполнять операцию '`flush`' после каждого хода, либо не будете соблюдать формат взаимодействия с программой жюри, вы можете получить любой вердикт.

Пример

стандартный ввод	стандартный вывод
2 00 1 0	Alice 2 2

Задача M. Доставщик пиццы (сложно)

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

В государстве находится n офисов, в каждом из которых работает множество сотрудников. И сотрудники каждого офиса могут загрустить! Доставщики пиццы об этом знают, и они не хотят потерять возможность поднять настроение офисным сотрудникам, продав им много больших пицц.

Сейчас доставщики пицц планируют построить самую большую пиццерию в государстве, однако они еще не определились с ее местоположением. Поскольку для доставщиков пицц каждый километр дается значительно тяжелее предыдущего, доставщики хотят найти такую точку в государстве, из которой можно было бы добраться до самого удаленного офиса за наименьшее возможное время.

Государство не обычное, а четырехмерное. Поверхностью планеты, на которой находится государство, является трехмерное пространство. Разумеется, все офисы расположены на данном трехмерном пространстве.

Расстояние между точками $(x_i,\ y_i,\ z_i)$ и $(x_j,\ y_j,\ z_j)$ вычисляется по формуле $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$. Пиццерию можно строить в любой точке, даже в каком-либо из офисов.

Экономика государства в опасности! Сообщите, где необходимо построить пиццерию, чтобы всем жилось хорошо!

Формат входных данных

В первой строке входного файла содержится целое число n — количество офисов ($1 \leqslant n \leqslant 2000$). Каждая из последующих n строк содержит информация о офисах. i-я из этих строк содержит три целых числа x_i, y_i, z_i — координаты i-го офиса ($-10^4 \leqslant x_i,\ y_i\ z_i \leqslant 10^4$). Никакие два офиса не совпадают.

Формат выходных данных

В первой строке выходного файла следует вывести три вещественных числа через пробел x_0, y_0, z_0 — координаты будущей пиццерии. Если существует несколько решений, то разрешается вывести любое. Ответ будет засчитан, если расстояние от данной точки до самого удаленного офиса будет отличаться от результата жюри не более чем на 10^{-6} по абсолютному или относительному значению.

Пример

стандартный ввод	стандартный вывод
5 4 -6 -8 3 -10 -3 3 5 2 -2 3 10 -1 -9 -8	-1.26216215 -2.89639639 0.94414414

Замечание

В примере максимальное расстояние от базы до какой-либо из планет примерно равняется 10.8314551386.

Задача N. Угадать количество делителей

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

Это интерактивная задача.

Загадано число $1 \leqslant X \leqslant 10^9$. Вам **не нужно** угадывать это число. Вам нужно **определить количество делителей** этого числа, и даже это вам **не нужно делать точно**: ваш ответ будет считаться верным, если его абсолютная погрешность не превышает 7 **или** его относительная погрешность не превышает 0.5. Формально, пусть ваш ответ равен ans, а количество делителей X равно d, тогда ваш ответ будет считаться правильным, если выполнено **хотя бы одно** из следующих двух условий:

- $|ans - d| \leqslant 7$
- $\frac{1}{2} \leqslant \frac{ans}{d} \leqslant 2$

Вы можете не более 22 раз сделать запрос. Запрос состоит из одного числа $1 \leqslant Q \leqslant 10^{18}$. В ответ на запрос вы получите $gcd(X, Q)$ — наибольший общий делитель X и Q.

Число X зафиксировано до всех запросов. Иными словами, **интерактор не является адаптивным**.

Назовём процесс отгадывания количества делителей числа X *игрой*. В рамках одного теста вам нужно будет сыграть T независимых игр, то есть отгадать количество делителей T раз для T независимых чисел X.

Формат входных данных

На первой строке записано одно целое число T ($1 \leqslant T \leqslant 100$) — количество игр.

Протокол взаимодействия

Чтобы сделать запрос, выведите строку вида «`? Q`» ($1 \leqslant Q \leqslant 10^{18}$). После запроса считайте одно число — $gcd(X, Q)$. Вы можете сделать не более 22 таких запросов в рамках одной игры.

Если вы считаете, что знаете количество делителей X с достаточной точностью, выведите ваш ответ в формате «`! ans`». ans должно быть целым числом. Если это последняя игра, то вы должны завершить выполнение программы, иначе вы должны начать следующую игру. Обратите внимание, что интерактор не выводит ничего в ответ на вывод ответа.

После вывода запроса или ответа не забудьте вывести перевод строки и сбросить буфер вывода. Для сброса буфера вывода используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Пример

стандартный ввод	стандартный вывод
2 1 1 1 1024 1048576 4194304	 ? 982306799268821872 ? 230856864650023977 ? 134690134760714371 ! 5 ? 1024 ? 1048576 ? 1073741824 ! 42

Замечание

Почему ограничение на запросы именно 22? Возможно, автор задачи — фанат Тейлор Свифт.

Рассмотрим пример из условия.

В первой игре загадано число $X = 998\,244\,353$. Было бы сложно это угадать, правда? Это число является простым, то есть количество его делителей равно 2. Решение сделало запросы с несколькими случайными числами, и ответы на все запросы оказались равны 1 (удивительно, что ни один из трёх запросов не оказался кратным $998\,244\,353$). Логично предположить, что у загаданного числа не очень много делителей, поэтому решение ответило 5. Почему бы и не 5. Этот ответ будет засчитан, так как $|5 - 2| = 3 \leqslant 7$.

Во второй игре загадано число $X = 4\,194\,304 = 2^{22}$, количество его делителей равно 23. Решение сделало запросы $1024 = 2^{10}$, $1\,048\,576 = 2^{20}$, $1\,073\,741\,824 = 2^{30}$ и получило ответы $1024 = 2^{10}$, $1\,048\,576 = 2^{20}$, $4\,194\,304 = 2^{22}$, соответственно. Затем решение окончательно запуталось и выдало ответ на Главный вопрос жизни, Вселенной и всего такого. Этот ответ будет засчитан, так как $\frac{1}{2} \leqslant \frac{42}{23} \leqslant 2$.

Задача O. All your base are belong to us

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

В 2021 году нашей эры началась война. Враг захватил все наши базы. Чтобы снова завладеть нашими базами, было решено обосновать штаб-квартиру. При том нужно расположить штаб-квартиру в таком месте, чтобы все базы были не очень далеко от нее. Поэтому мы решили выбрать такое место, чтобы минимизировать суммарное расстояние от штаб-квартиры до K самых далеких от нее баз.

Все базы являются точками на плоскости. Мы можем расположить штаб-квартиру в любой точке плоскости (даже если у этой точки не целые координаты).

От вас требуется определить оптимальное расположение штаб-квартиры.

Формат входных данных

В первой строке записаны два числа N и K. Число N обозначает количество баз ($1 \leqslant N \leqslant 200$). Число K отвечает за то, расстояние до скольки баз мы учитываем в сумме ($1 \leqslant K \leqslant N$).

В каждой из следующих N строк записаны целые числа x_i и y_i — координаты i-й базы ($-1000 \leqslant x_i, y_i \leqslant 1000$).

Формат выходных данных

Выведите минимальную сумму расстояний от штаб-квартиры до K самых далеких от нее баз. Ответ считается корректным, если абсолютная или относительная погрешность не превосходит 10^{-4}.

Примеры

стандартный ввод	стандартный вывод
3 1 0 1 1 0 1 1	0.70711
6 3 1 1 2 1 3 2 5 3 8 5 13 8	17.50426
9 3 573 -50 -256 158 -751 14 314 207 293 567 59 -340 -243 -22 -268 432 -91 -192	1841.20904

Задача P. Теория Игр

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

Костя отыскал у себя дома N кубиков и принялся играть в одну увлекательную игру: он захотел воспользоваться находкой и построить максимально высокую башню!

У Кости есть большой стол, на котором изначально кубиков нет. Строительство башни состоит из последовательности действий Кости. Во время каждого действия Костя берет очередной кубик и ставит его сверху на уже построенную башню, либо на стол, если ранее на столе кубиков не было. После каждого действия Костя выписывает на лист количество кубиков в получившейся башне. Когда у Кости кубиков не останется, он закончит игру.

Его другу Мише такая игра показалась скучной, и он решил в неё вмешаться. После того, как Костя совершит очередное действие и запишет на лист количество кубиков в новой башне, Миша будет делать одно из двух действий:

- Не вмешиваться в игру и позволить Косте продолжать строительство башни.
- Разрушить башню, которая на данный момент находится на столе и забрать все кубики, из которых состояла башня, себе. Если Миша забрал себе некоторые кубики, он уже ни за что не отдаст их Косте.

Если Миша вмешается в игру после очередного действия Кости, чтобы сильно не расстраиваться и не обижаться на друга, Костя запишет себе на лист поощрительное число C. После этого он продолжит свою игру, начав строить башню заново уже из оставшихся кубиков.

Миша не любит большие числа и хочет, чтобы сумма всех записей, сделанных Костей во время игры была как можно меньше.

Помогите Мише найти сумму чисел, записанных Костей, в конце игры, если он будет действовать оптимально.

Формат входных данных

В единственной строке через пробел записаны два целых числа N и C ($1 \leqslant N, C \leqslant 10^9$) — количество кубиков у Кости и поощрительное число, соответственно.

Формат выходных данных

Выведите одно целое число — минимальную сумму чисел, записанных Костей на лист.

Примеры

стандартный ввод	стандартный вывод
3 3	6
5 9	15
1000000 1	1999999

Замечание

В первом примере оптимальной стратегией Миши является не вмешиваться в игру и позволить Косте построить башню, состоящую из трех кубиков. Тогда сумма записанных чисел будет равна: $1 + 2 + 3 = 6$.

Во втором примере также следует не мешать Косте и позволить построить башню из пяти кубиков.

В третьем примере оптимальным является разрушать башню после каждого действия Кости, кроме последнего. Таким образом, сумма записанных чисел будет равна $1\,000\,000 + 1 \cdot 999\,999$.

Задача Q. Дом для червячка

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

Это интерактивная задача.

Вы ищете место в земле, куда вы бы могли поместить свое любимое домашнее животное — червячка Максимуса! В качестве зоны поисков вы выбрали клочок земли размера $N \times M \times K$. Выбранный клочок земли поделен на $N \cdot M \cdot K$ единичных кубических клеток размера $1 \times 1 \times 1$. Каждая клетка имеет свои координаты (x, y, z), где $1 \leqslant x \leqslant N$, $1 \leqslant y \leqslant M$ и $1 \leqslant z \leqslant K$. Также для каждой клетки при помощи специального сканера может быть определена ее влажность $h(x, y, z)$, которая является целым числом от 1 до 10^9. При желании вы можете измерить влажность любой клетки выбранного клочка земли.

Максимус очень любит влажную почву, поэтому вы хотите выбрать для него такую клетку земли, влажность которой не меньше, чем влажность всех шести соседних клеток. В противном случае червячок может не согласиться с вашим выбором и уползти в соседние клетки, после чего найти его будет крайне сложно. Формально, от вас требуется найти клетку (x, y, z), которая является *локальным максимумом* влажности, то есть для нее должно выполняться условие:

$h(x, y, z) \geqslant \max(h(x-1, y, z), h(x+1, y, z), h(x, y-1, z), h(x, y+1, z), h(x, y, z-1), h(x, y, z+1))$.

Для удобства будем считать влажность клеток, находящихся за пределами выбранного участка, равной нулю.

Количество клеток в выбранном клочке земли может быть достаточно большим, и, разумеется, вы не хотите потратить слишком много времени, чтобы измерить влажность всех клеток. Поэтому перед вами встала непростая задача — измерить влажность не слишком большого количества клеток и найти клетку (x, y, z), обладающую нужными свойствами.

Протокол взаимодействия

В начале взаимодействия с программой жюри вы должны считать четыре целых числа N, M, K и Q — размеры клочка земли, а также максимальное количество измерений влажности, которое вам разрешено сделать.

После этого вы можете не более, чем Q раз выяснить влажность почвы в некоторой клетке, выведя запрос вида: «`? x y z`» ($1 \leqslant x \leqslant N$, $1 \leqslant y \leqslant M$, $1 \leqslant z \leqslant K$). После вывода запроса вы должны считать ответ программы жюри — одно целое число в диапазоне от 1 до 10^9 — влажность почвы в клетке (x, y, z). В случае, если вы превысили количество допустимых запросов, в качестве ответа вы получите число -1. В этом случае вы должны немедленно завершить выполнение программы.

В случае, если вы нашли клетку (x, y, z), являющуюся подходящей для размещения в ней червячка, вы должны вывести запрос вида: «`! x y z`» ($1 \leqslant x \leqslant N$, $1 \leqslant y \leqslant M$, $1 \leqslant z \leqslant K$) и немедленно завершить выполнение программы.

После каждого запроса, в том числе после последнего, вы должны выполнять операцию `flush`. Примеры выполнения данной операции для разных языков программирования приведены ниже:

- Java: `System.out.println()` выполняет операцию `flush` автоматически.
- Python: `print()` выполняет операцию `flush` автоматически.
- C++: `cout << endl` выполняет операцию `flush` автоматически. В случае использования `printf`, выполняйте операцию `fflush(stdout)`.
- Pascal: `Flush(Output)`.

В случае несоблюдения описанного протокола ваше решение может получить любой вердикт.

Гарантируется, что программа жюри **не является адаптивной**. Это значит, что каждый тест зафиксирован до запуска решения и не может меняться в процессе общения решения и программы жюри.

Система оценки

Баллы за каждую группу выдаются только в случае, если все тесты группы успешно пройдены. Ограничения на N, M, K и Q записаны в таблице ниже. Группы оцениваются независимо друг от друга.

Группа	Баллы	N	M	K	Q
0	0	Тесты из условия			
1	10	$N = 1\,000\,000$	$M = 1$	$K = 1$	$Q = 10\,000$
2	22	$N = 1\,000\,000$	$M = 1$	$K = 1$	$Q = 35$
3	12	$N = 200$	$M = 200$	$K = 1$	$Q = 4\,000$
4	19	$N = 1\,000$	$M = 1\,000$	$K = 1$	$Q = 3\,500$
5	14	$N = 100$	$M = 100$	$K = 100$	$Q = 100\,000$
6	23	$N = 500$	$M = 500$	$K = 500$	$Q = 150\,000$

Пример

стандартный ввод	стандартный вывод
3 1 1 3 13 14 10	 ? 3 1 1 ? 2 1 1 ? 1 1 1 ! 2 1 1

Задача R. Дороги в Ельце

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

Елец, Елец, не город, а ...

Фольклор

Это интерактивная задача.

Как хорошо известно, в городе Елец за его полуторатысячелетнюю историю ни разу не ремонтировали дороги. И только недавно руководство города отремонтировало некоторые из них.

Известно, что всего в Ельце есть n перекрестков и m дорог, перемещаться по которым можно в обе стороны, пронумерованных целыми числами от 1 до m. i-я дорога соединяет перекрестки с номерами a_i и b_i.

Среди всех m дорог было отремонтировано некоторое подмножество дорог, но вам не известно, какое именно. Единственная информация, которую вы смогли получить от дорожных служб города, это то, что от любого перекрестка можно доехать до любого другого, двигаясь только по отремонтированным дорогам.

Вы — молодой предприниматель, решили организовать службу доставки свежего сырого мяса в Ельце (в самом городе такое мясо называют «стейками», оно пользуется большой популярностью у местных жителей). Вы уже набрали штат курьеров, однако курьеры готовы перемещаться только по отремонтированным дорогам. Теперь вам предстоит выяснить, какие дороги уже отремонтированы.

Городская администрация предоставила вам город на некоторое время, поэтому вы можете делать действия одного из трех типов:

1. Заблокировать дорогу с номером x. В этом случае перемещение по дороге для курьеров будет запрещено. Исходно все дороги разблокированы.

2. Разблокировать дорогу с номером x. В этом случае курьеры смогут двигаться по дороге x, если она отремонтирована.

3. Попробовать доставить заказ на перекресток с номером y. В этом случае один из ваших курьеров начнет двигаться от неизвестного вам перекрестка s и доставит заказ на перекресток с номером y в том случае, если существует путь по разблокированным отремонтированным дорогам от перекрестка s до перекрестка y. При этом гарантируется, что перекресток s будет выбран заранее.

К сожалению, город предоставлен в ваше полное распоряжение не надолго, поэтому вы можете сделать не более $100 \cdot m$ запросов.

Протокол взаимодействия

Ваша программа будет взаимодействовать с программой жюри с использованием стандартных потоков ввода и вывода. Каждое взаимодействие будет состоять из решения задачи для нескольких наборов входных данных.

Сначала ваша программа должна считать целое число t ($1 \leqslant t \leqslant 1\,000$) — количество наборов входных данных в рамках одного взаимодействия с программой жюри. Затем t раз необходимо выполнить взаимодействие по решению задачи для набора входных данных.

Рассмотрим протокол взаимодействия для одного набора входных данных.

Сначала ваша программа должна считать данные в следующем формате.

Первая строка содержит два целых числа n и m ($2 \leqslant n \leqslant 2\,000$, $n - 1 \leqslant m \leqslant 2\,000$) — число перекрестков и дорог в Ельце.

Каждая из следующих m строк содержит описание одной дороги. i-я из этих строк содержит два целых числа a_i и b_i ($1 \leqslant a_i, b_i \leqslant n$) — номера перекрестков, которые соединяет i-я дорога. Гарантируется, что никакая дорога не соединяет перекресток сам с собой. При этом возможно, что между парой различных перекрестков есть несколько дорог.

После того, как вы считали описание набора входных данных, вы можете задавать запросы. Запросы могут быть трех типов:

1. «`- x`» ($1 \leqslant x \leqslant m$). В этом случае дорога с номером x блокируется, если она ещё не была заблокирована.

2. «`+ x`» ($1 \leqslant x \leqslant m$). В этом случае дорога с номером x разблокируется. **Обратите внимание**, что дорога x должна быть заблокирована ранее. Исходно все дороги разблокированы.

3. «`? y`» ($1 \leqslant y \leqslant n$). В этом случае программа жюри выбирает некоторый город s. В случае, если от города s до города y можно добраться по разблокированным отремонтированным дорогам, программа жюри выведет 1, иначе программа жюри выведет 0. **Обратите внимание**, что город s будет выбран до получения информации о городе y, однако при выборе города s могут учитываться ваши предыдущие запросы.

Всего вы можете задать не более $100 \cdot m$ запросов для каждого набора входных данных.

После того, как вы нашли все отремонтированные дороги, выведите «`! c1 c2 c3 ... cm`», где c_i равно 1, если дорога i отремонтирована, и 0, если дорога не отремонтирована. Этот вывод не будет считаться в общем числе запросов. На это программа жюри выведет 1, если ваш ответ является правильным, и 0 в противном случае. В случае, если ответ оказался не правильным, ваша программа должна немедленно завершить работу.

Обратите внимание, что вам не обязательно разблокировать все дороги на момент вывода ответа. Гарантируется, что все отремонтированные дороги зафиксированы изначально и не будут меняться программой жюри в зависимости от запросов.

Гарантируется, что суммарно по всем наборам входных данных в одном тесте, сумма n и сумма m не превосходят 2 000.

Если вы используете «`cout << ... << endl`» в C++, «`System.out.println`» в Java, «`print`» в Python, «`writeln`» в Pascal, то сброс потока вывода происходит автоматически, дополнительно ничего делать не требуется.

Если вы используете другой способ вывода, рекомендуется делать сброс буфера потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае. Для сброса буфера потока вывода можно использовать «`fflush(stdout)`» в C++, «`flush(output)`» в Pascal, «`System.out.flush()`» в Java, «`sys.stdout.flush()`» в Python.

Пример

стандартный ввод	стандартный вывод
2 2 2 1 2 2 1	
	- 1 ? 1
1	
	? 2
0	
	- 2 + 1 ? 1
1	
	! 1 0
1 3 3 1 2 2 3 3 1	
	- 1 ? 2
1	
	? 1
1	
	- 2 ? 3
1	
	? 3
0	
	+ 1 ? 3
1	
	? 2
1	
	? 1
1	
	! 1 1 1
1	

Замечание

В первом наборе входных данных дорога 1 отремонтирована, а дорога 2 — нет. Для первого запроса доставки в качестве s был выбран перекресток 1, поэтому путь от перекрестка 1 до 1 есть. Для второго запроса доставки в качестве s был выбран перекресток 1. Так как в городе заблокирована единственная отремонтированная дорога, то пути между перекрестками 1 и 2 нет. Для третьего запроса доставки в качестве s был выбран перекресток 2, путь между перекрестками 2 и 1 есть по дороге 1, которая отремонтирована и разблокирована.

Во втором наборе входных данных для запросов доставки в качестве стартовых перекрестков были выбраны перекрестки 1, 3, 1, 2, 2, 3, 1.

Задача S. Угадайка

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

Это интерактивная задача.

Ваша задача — написать программу, которая угадывает загаданное число, задавая вопросы. Известно, что было загадано целое неотрицательное число, меньшее чем 10^{18}.

Пусть x — загаданное число. За один вопрос вы можете выбрать некоторое целое неотрицательное число a, после чего в качестве ответа на вопрос вы получите сумму цифр числа $(x + a)$. Здесь сумма цифр — это сумма всех цифр в десятичной записи числа. Например, сумма цифр числа 4096 равна $4 + 0 + 9 + 6 = 19$.

Протокол взаимодействия

Вы должны выводить вопросы в поток стандартного вывода и считывать ответы из потока стандартного ввода. Вы можете задавать много вопросов. Когда вы будете уверены, что знаете загаданное число, вы должны дать ваш ответ.

Для того, чтобы задать вопрос, вы должны вывести следующую строку:

`query` a

Здесь a — это целое число от 0 до $10^{18} - 1$, включительно. В качестве ответа вы получите сумму цифр числа $(x + a)$, где x — загаданное число.

Чтобы дать ответ, вы должны вывести следующую строку:

`answer` y

Здесь y — это загаданное число от 0 до $10^{18} - 1$, включительно, определенное вами. Вы можете дать ответ лишь один раз. Вы можете задать не более, чем 75 вопросов, прежде чем дать ответ.

Пример

стандартный ввод	стандартный вывод
15 1	query 3 query 25 answer 75

Замечание

В случае не соблюдения протокола взаимодействия ваше решение может получить произвольный вердикт. Не забывайте выполнять операцию `flush` после каждого вопроса, а также после того, как вы дадите ответ.