

Задача А. 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 ($1 \leq n \leq 10^9$). Затем протокол общения следующий: требуется вывести в стандартный поток вывода одну строку, содержащую целое число — 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	
	3
-1	
	1
1	
	2
0	

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 В. 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, \dots, a_N . Маша недавно изучила алгоритмы сортировки, поэтому она выписала свои числа в неубывающем порядке, то есть $a_1 \leq a_2 \leq \dots \leq a_N$.

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 $1 \leq a_i \leq 10^9$.

Протокол взаимодействия

В начале ваше решение должно считать число N ($1 \leq N \leq 10^6$) — 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, $1 \leq i \leq N$). 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	? 2
1	? 3
4	? 4
5	? 5
5	! 4

Замечание

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

Задача С. 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 $1 \leq t \leq n$, что для любого $1 < i \leq t$ $s_{i-1} < s_i$ и для любого $t < j \leq n$ $s_{j-1} > s_j$. Чтобы подтвердить свою правоту, Великому сквайру нужно узнать, в какой момент времени площадь владений Стиня в точности равнялась k , и он просит вас помочь ему. Вы можете по моменту времени i узнать размер владений сквайра Стиня 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 успехе, и гарантирует вам, что искомое s_i существует.

Протокол взаимодействия

Изначально вам заданы два числа n, k — количество записей о владениях Стиня в архиве и значение площади, которое интересует Файфа ($2 \leq n \leq 2 \cdot 10^5, 0 \leq k \leq 10^9$). 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, $0 \leq s_i \leq 10^9$. 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	? 5
2	? 4
8	? 3
10	? 1
1	? 2
3	! 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 ($1 \leq k \leq n \leq 10^6$).

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

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 Е. 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 загрузить! 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 ($1 \leq n \leq 100$). 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 ($-10^4 \leq x_i, y_i, z_i \leq 10^4$). 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	-1.2621406320 -2.8964322792 0.9441692622
4 -6 -8	
3 -10 -3	
3 5 2	
-2 3 10	
-1 -9 -8	

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

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

Формат входных данных

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

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:

- «? l r» — узнать сумму чисел на отрезке массива с l -го по r -й элемент включительно;
- «!» — 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 ($1 \leq n \leq 10^4$). 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 ($1 \leq l \leq r \leq n$). После каждого запроса должен следовать перевод строки. При несоблюдении вашей программой формата запросов, ваше решение может получить произвольный вердикт (отличный от ОК).

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 ($|s| \leq 10^{18}; 1 \leq l_b \leq r_b \leq n$).

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 PL, считав из ввода значения «-1 -1 -1», ваша программа должна завершить свою работу с нулевым кодом возврата.

Пример

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

Задача Н. 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 убьёт 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

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

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 ($1 \leq g \leq 1000$).

Для каждого теста зафиксировано число q ($6 \leq q \leq 60$) — 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 ($1 \leq t \leq 10^{18}$), для которого требуется узнать значение выражения

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

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 ($1 \leq t \leq 10^{18}$). 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	X 2
1 2	X 3
3 10	X 5
1 5	X 4
1 5	X 6
1 10	X 10
1 10	X 11
0 1	N 10
Correct	X 1
1 1	X 2
0 1	N 1
Correct	

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 ($1 \leq P \leq B$), 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 $1, 11, 21, 31, 41, \dots$, 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 ($1 \leq T \leq 100$) — 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	? 1
0	? 2
0	? 3
0	? 12
1	
Y	! 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 мандарины, 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:

- ? $x_1 y_1 x_2 y_2$ ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq n$) — 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) ;
- ! $x_1 y_1 x_2 y_2$ ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq n$) — когда Глеб уверен, что он точно знает, где находятся мандарины, 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 ($1 \leq n \leq 2 \cdot 10^9$) — 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	? 1 1 4 4
6	? 1 3 4 4
6	? 2 3 4 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	$n \leq 10$	$q = 10\,000$	
2	13 – 23	20	$n \leq 100$	$q = 10\,000$	
3	24 – 34	20	$n \leq 10\,000$	$q = 20\,000$	
4	35 – 45	25	$n \leq 2 \cdot 10^9$	$q = 128$	
5	46 – ∞	25	$n \leq 2 \cdot 10^9$	$q = 64$	

Задача К. 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, \dots, a_n , которое вычисляется по формуле:

$$(\dots(((a_0 \text{ op}_1 a_1) \text{ op}_2 a_2) \text{ op}_3 a_3) \dots \text{ op}_n a_n) \bmod 10^9 + 7,$$

где $\text{op}_1, \text{op}_2, \dots, \text{op}_n$ — это арифметические операции, каждая из которых является либо операцией «+» (сложение), либо операцией «×» (умножение), а a_0, a_1, \dots, a_n — переменные.

Например, если значения переменных равны $(a_0, a_1, a_2) = (1, 1, 2)$, а операции равны $(\text{op}_1, \text{op}_2) = (+, \times)$, 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 $\text{op}_1, \text{op}_2, \dots, \text{op}_n$.

Протокол взаимодействия

В начале взаимодействия с программой жюри вы должны считать целое число n ($1 \leq n \leq 4000$) — количество операторов в формуле.

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

Когда вы определите, чему равны $\text{op}_1, \text{op}_2, \dots, \text{op}_n$, вы должны вывести запрос вида: «! s », где s — строка, состоящая из n символов + или × (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 выведете 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	? 1 1 2
4	? 1 1 3
6	! +x
10	? 1 1 1 1 1 1 1 1 1 1
5	? 0 4 2 4 2 4 2 4 2 4 2
6224	? 1 2 3 4 5 6 7 8 9 10 11
640750	! ++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 ($1 \leq n \leq 15$) — 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 ($0 \leq p \leq n$).

- $p = 0$ символизирует, что вы сдаётесь. $1 \leq p \leq n$ — 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 ($-1 \leq p \leq n$).

- $p = 0$ 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. $1 \leq p \leq n$ — 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	Alice
00	2
1	2
0	

Задача М. Доставка 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! Доставка 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 ($1 \leq n \leq 2000$). 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 ($-10^4 \leq x_i, y_i, z_i \leq 10^4$). 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	-1.26216215 -2.89639639 0.94414414
4 -6 -8	
3 -10 -3	
3 5 2	
-2 3 10	
-1 -9 -8	

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 $1 \leq X \leq 10^9$. 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:

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

Вы можете не более 22 раз сделать запрос. Запрос состоит из одного числа $1 \leq Q \leq 10^{18}$. 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 ($1 \leq T \leq 100$) — количество игр.

Протокол взаимодействия

Чтобы сделать запрос, выведите строку вида «? Q» ($1 \leq Q \leq 10^{18}$). 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	? 982306799268821872
1	? 230856864650023977
1	? 134690134760714371
1	! 5 ? 1024
1024	? 1048576
1048576	? 1073741824
4194304	! 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 $|5 - 2| = 3 \leq 7$.

Во второй игре загадано число $X = 4\,194\,304 = 2^{22}$, количество его делителей равно 23. Решение сделало запросы $1024 = 2^{10}$, $1048576 = 2^{20}$, $1073741824 = 2^{30}$ и получило ответы $1024 = 2^{10}$, $1048576 = 2^{20}$, $4194304 = 2^{22}$, соответственно. Затем решение окончательно запуталось и выдало ответ на Главный вопрос жизни, Вселенной и всего такого. Этот ответ будет засчитан, так как $\frac{1}{2} \leq \frac{42}{23} \leq 2$.