

## 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, задав не более чем за 30 вопросов. Для этого программа участника сообщает свои догадки программе жюри, а программа жюри отвечает, является ли загаданное число большим, меньшим или равным сделанной догадке.

### Протокол взаимодействия

Сначала необходимо прочитать из стандартного потока ввода число  $n$  ( $1 \leq n \leq 30$ ). 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.

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++, вывод `endl` автоматически делает flush;
- `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
5	
	3
-1	
	1
1	
	2
0	

## Задача В. 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 С. 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 D. 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 интересуется Файф ( $2 \leq n \leq 2 \cdot 10^5, 0 \leq k \leq 10^9$ ). 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

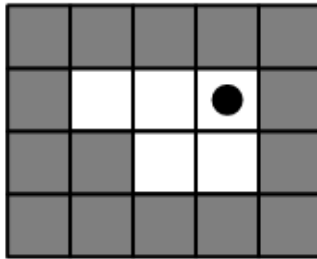
В данном примере записи о владениях выглядели так: 1, 3, 10, 8, 2.

## Задача E. Вслепую по лабиринту

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

Ваша цель — написать программу, управляющую роботом, идущим вслепую по лабиринту.

Лабиринт состоит из  $n$  на  $m$  ( $1 \leq n, m \leq 30$ ) клеток. Каждая из клеток может быть свободной или заблокированной (непроходимой). Все клетки на границе лабиринта непроходимые. Робот начинает работу в свободной клетке, он может переместиться на юг, запад, север или восток в свободную клетку. Робот не имеет оптических сенсоров, только сенсор удара, так что при попытке перемещения в заблокированную клетку сработает сенсор и робот останется в той же клетке.



Робот должен побывать во всех проходимых клетках лабиринта. Из начальной клетки гарантированно можно попасть во все достижимые клетки лабиринта.

### Формат входных данных

Каждая строка выходных данных должна представлять собой команду для робота. Это должна быть одна из пяти возможных строк: `SOUTH`, `WEST`, `NORTH`, `EAST` или `DONE`. Строка `DONE` должна быть напечатана после посещения роботом всех проходимых клеток. После вывода `DONE` программа должна завершать свою работу. Необходимо очищать выходной буфер после вывода каждой команды (`flush`).

### Формат выходных данных

Каждая строка входных данных представляет собой ответ на действие робота. Это может быть строка `EMPTY`, если робот успешно переместился в заданном направлении, или строка `BLOCKED`, если робот не смог переместиться из-за того, что клетка, в которую он хотел попасть, непроходима.

### Протокол взаимодействия

Программа должна выводить на стандартный вывод одну строку с действием робота и ждать строки в стандартном вводе с ответом, затем выводить очередную строку с действием и считывать ответ и так далее до тех пор, пока все проходимые клетки лабиринта не будут посещены. Программа должна завершать работу только когда все проходимые клетки будут посещены. Проходимые клетки могут быть посещены несколько раз. Допустимо передвигаться даже если все проходимые клетки уже посещены.

## Пример

стандартный ввод	стандартный вывод
BLOCKED	NORTH
BLOCKED	EAST
EMPTY	SOUTH
BLOCKED	EAST
BLOCKED	SOUTH
EMPTY	WEST
BLOCKED	SOUTH
BLOCKED	WEST
EMPTY	NORTH
EMPTY	WEST
BLOCKED	WEST
BLOCKED	NORTH
EMPTY	EAST
BLOCKED	NORTH
	DONE

## Задача F. 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:

- ?  $x_1 y_1 x_2 y_2$  ( $1 \leq x_1 \leq x_2 \leq n$ ,  $1 \leq y_1 \leq y_2 \leq n$ ) — 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 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  $n$  ( $1 \leq n \leq 2 \cdot 10^9$ ) — 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.



## Задача G. 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 вы 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 Н. Последний рубеж

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

### Это интерактивная задача.

Однажды общительный программист Павел позвал своих друзей на квест. Ребята с лёгкостью решали головоломки и продвигались вперёд. И вот им осталось решить последнюю загадку перед тем, как получить долгожданный приз.

Загадка состоит в том, что перед ребятами находится дверь с  $N$  замками, которую нужно открыть. Некоторые замки открыты, а некоторые закрыты. Ребята не знают, какие замки уже открыты, однако, потратив некоторое время на изучение одного конкретного замка, они могут определить, открыт он или нет. Рядом с дверью висит табличка, на которой написано, что самый левый замок открыт, а самый правый закрыт.

В процессе выполнения предыдущих заданий ребята выяснили, как открыть дверь. Пронумеруем замки слева направо числами от 1 до  $N$ . Тогда для того, чтобы открыть дверь, ребятам нужно найти замок с номером  $i < N$ , такой что замок  $i$  открыт, а замок  $i + 1$  закрыт.

Как уже было сказано, для того, чтобы определить, является ли  $i$ -й замок открытым, им нужно подробно его осмотреть, потратив на это некоторое время. Так как у ребят осталось немного времени для выполнения последнего задания, они могут подробно осмотреть не более  $Q$  замков.

Помогите ребятам открыть дверь.

### Протокол взаимодействия

В начале на вход программе подаётся одно целое число  $N$  ( $2 \leq N \leq 10^{18}$ ).

После этого вы можете делать запросы вида  $? i$ , означающие, что ребята подробно осматривают замок с номером  $i$ . В ответ на подобный запрос вы получите число 0, означающее, что замок закрыт, или число 1 в противном случае.

Если вы нашли ответ, вы должны сделать запрос вида  $! i$ , означающий, что вы считаете, что замок  $i$  открыт, а замок  $i + 1$  закрыт. После этого запроса вы должны завершить работу программы.

Считается, что в начале вы знаете состояние замков 1 и  $N$ .

Вы должны сделать не более  $Q$  запросов первого типа. В случае превышения этого ограничения ваше решение получит вердикт «Неправильный ответ».

В случае нарушения каких-либо правил взаимодействия с программой-интерактором, ваше решение может получить любой вердикт.

После каждого запроса, в том числе после запроса второго типа, 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;
- `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.

## Примеры

стандартный ввод	стандартный вывод
3 0	? 2 ! 1
3 1	? 2 ! 2
5 0 0 1	? 2 ? ? ! ? ? ? ! ? ? ? !