

Задача А. Наименьшее количество инверсий

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

Это задача с двойным запуском. У Вани была перестановка p размера n : последовательность из чисел $1, 2, \dots, n$, в которой каждое число встречается один раз. Ваня хотел бы узнать число x — количество инверсий в ней, то есть количество пар позиций (i, j) , для которых $1 \leq i < j \leq n$ и при этом $p_i > p_j$. Важное дополнение: Ваня не успел сам посчитать x , ибо судьба перестановки непроста. Сначала в гости к Ване зашёл Влад и перенёс эту перестановку на ленту, концы которой склеены так, что после последнего элемента следует первый. Теперь неизвестно, с какого из n элементов начиналась перестановка. Тем не менее, Ваня уверен, что число x — мини-мальное из тех, что могут получиться, если разрезать ленту по какой-то из n линий между соседними числами. Затем в гости к Ване зашёл Никита и разрезал эту ленту на два непустых куска: в каждом куске получилось от 1 до $n - 1$ чисел. Теперь Ваня готов сначала показать вам первый кусок, а потом спрятать его, пока с ним ещё что-нибудь не случилось, и показать второй кусок. Несмотря на все трудности, помогите Ване узнать число x .

Формат входных данных

В этой задаче ваше решение будет запущено на каждом тесте два раза. При первом запуске решение получает первый кусок ленты. В первой строке записано слово «first». Во второй строке заданы два целых числа n и s — длина перестановки и размер первого куска ($1 \leq s < n \leq 300000$). В третьей строке заданы s чисел p_1, p_2, \dots, p_s — первый кусок перестановки.

При втором запуске решение получает второй кусок ленты, а также строку, выведенную при первом запуске. В первой строке записано слово «second». Во второй строке заданы два целых числа n и t — длина перестановки, та же, что при первом запуске, и размер второго куска. В третьей строке заданы t чисел q_1, q_2, \dots, q_t — второй кусок перестановки. Гарантируется, что первый и второй кусок вместе образуют перестановку длины n . В четвёртой строке приведена строка h , выведенная при первом запуске. В каждом тесте вся перестановка зафиксирована заранее и не меняется между запусками.

Формат выходных данных

После первого запуска введите строку h , содержащую от 1 до 100 произвольных символов с ASCII-кодами от 32 до 126 включительно.

После второго запуска выведите одно целое число x : минимально возможное количество инверсий в перестановке, если она могла начинаться с любой позиции на ленте.

Примеры

стандартный ввод	стандартный вывод
first 5 3 1 5 4	!@# 1 5 4 #@!
second 5 2 3 2 !@# 1 5 4 #@!	4

Замечание

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. В примерах показаны два запуска какого-то решения на каком-то тесте.

Задача В. Пересылка через репитер

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

Это задача с двойным запуском. Вика хочет передать Серёже строку из n двоичных цифр. Для передачи используется репитер. Вика может послать не более $n + 100$ двоичных цифр на передачу. Серёжа должен, получив эти цифры, восстановить исходную строку. Важное дополнение: репитер выдаёт не передаваемую строку, а какой-то её циклический сдвиг. Формально, если передаваемая Викой строка равна $t_1t_2 \dots t_m$, то репитер выбирает произвольную позицию p от 1 до m , после чего выдаёт Серёже строку $t_pt_{p+1} \dots t_mt_1t_2 \dots t_{p-1}$. Помогите Вике и Серёже договориться о передаче так, чтобы Серёжа, получив строку от репитера, мог восстановить исходную строку.

Формат входных данных

В этой задаче ваше решение будет запущено на каждом тесте два раза. При первом запуске решение действует за Вику. В первой строке записано слово «send». Во второй строке задано целое число n — длина исходной строки ($1 \leq n \leq 450000$). В третьей строке задана исходная строка s , состоящая из n двоичных цифр.

При втором запуске решение действует за Серёжу. В первой строке записано слово «receive». Во второй строке задано целое число m — длина строки, переданной в первом запуске. В третьей строке задан циклический сдвиг переданной строки.

Формат выходных данных

После первого запуска выведите строку из двоичных цифр, которую следует передать. Количество цифр должно быть от 1 до $n + 100$ включительно.

После второго запуска выведите исходную строку s .

Примеры

стандартный ввод	стандартный вывод
send 5 01001	000000100111111
receive 15 011111100000010	01001

Замечание

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. В примерах показаны два запуска какого-то решения на каком-то тесте.

Задача С. Программирование квадрокоптеров

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

Школьники готовятся к участию в соревновании по программированию квадрокоптеров. Квадрокоптер, который используется в соревновании, может выполнять две команды: подняться вверх на 1 метр и опуститься вниз на 1 метр. Команда подъёма обозначается символом «(», а команда спуска — символом «)».

Программа для квадрокоптера представляет собой последовательность команд. Программа считается корректной, если, начав её исполнение на уровне земли и выполнив последовательно все команды, квадрокоптер снова оказывается на уровне земли. При этом в процессе выполнения программы квадрокоптер не должен пытаться опуститься ниже уровня земли.

Например, следующие программы являются корректными: «()()», «(((())»). Программа «(((» не является корректной, поскольку квадрокоптер завершает её выполнение на высоте 3 метра над уровнем земли, программа «()» также не является корректной, поскольку при выполнении третьей команды квадрокоптер пытается опуститься ниже уровня земли.

Участник соревнования написал корректную программу для квадрокоптера, состоящую из n команд, пронумерованных от 1 до n . Он загрузил её в память квадрокоптера для демонстрации во время соревнования. К сожалению, после загрузки программы в память квадрокоптера участник случайно удалил её на своём компьютере, а квадрокоптер не позволяет выгрузить программу из своей памяти.

К счастью, квадрокоптер поддерживает специальный режим отладки программы. В этом режиме квадрокоптер с загруженной в него программой может отвечать на специальные запросы. Каждый запрос представляет собой два целых числа: l и r , $1 \leq l \leq r \leq n$. В ответ на запрос квадрокоптер сообщает, является ли фрагмент загруженной в него программы, состоящий из команд с l -й по r -ю включительно, корректной программой для квадрокоптера, либо нет. Участник хочет с помощью режима отладки восстановить загруженную в квадрокоптер программу.

Требуется написать программу-решение, которая взаимодействует с программой жюри, моделирующей режим отладки квадрокоптера, и в итоге восстанавливает загруженную в квадрокоптер программу.

Формат входных данных

Это интерактивная задача.

Сначала на вход подаётся целое число n — количество команд в программе квадрокоптера ($2 \leq n \leq 50000$).

Для каждого теста жюри зафиксировано число k — максимальное количество запросов. Гарантируется, что k запросов достаточно, чтобы решить задачу. Это число не сообщается программе-решению. Ограничения k в различных подзадачах приведены в таблице системы оценивания. Если программа-решение делает более k запросов к программе жюри, то на этом тесте она получает в качестве результата тестирования «Неверный ответ».

Чтобы сделать запрос, программа-решение должна вывести строку вида «? $l r$ », где l и r — целые положительные числа, задающие фрагмент программы квадрокоптера ($1 \leq l \leq r \leq n$).

В ответ на запрос программы-решения программа жюри подаёт ей на вход либо строку «Yes», либо строку «No», в зависимости от того, является ли запрошенный фрагмент программы квадрокоптера корректной программой.

Если программа-решение определила ответ на задачу, то она должна вывести строку «! $c_1 c_2 \dots c_n$ », где символ c_i задаёт i -ю команду в программе квадрокоптера и равен либо «(», либо «)».

После этого программа-решение должна завершиться.

Гарантируется, что в каждом тесте программа в памяти квадрокоптера является фиксированной корректной программой, которая не меняется в зависимости от запросов, произведённых программой-решением.

В тестах $k = 100000$, то есть, разрешается произвести не более 100000 запросов.

Формат выходных данных

Примеры

стандартный ввод	стандартный вывод
4 <ожидание ответа> Yes <ожидание ответа> No <ожидание ответа> Yes <ожидание ответа> Yes <ожидание ответа>	<ожидание ввода> ? 1 4 <ожидание ввода> ? 1 3 <ожидание ввода> ? 1 2 <ожидание ввода> ? 3 4 <ожидание ввода> ! ()
6 <ожидание ответа> Yes <ожидание ответа> No <ожидание ответа> Yes <ожидание ответа>	<ожидание ввода> ? 3 4 <ожидание ввода> ? 1 2 <ожидание ввода> ? 2 5 <ожидание ввода> ! (())

Задача D. 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`.

Пример

стандартный ввод	стандартный вывод
4	\medskip
\medskip	? 1 1 4 4
6	\medskip
\medskip	? 1 3 4 4
6	\medskip
\medskip	? 2 3 4 4
4	\medskip
	! 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 E. The Cake is a Lie

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

GLaDOS приготовила для Челл новое, последнее испытание! В его рамках ей предстоит найти торт. Но всё не так-то просто — у неё отобрали порталную пушку и завязали глаза. Единственное, что Челл известно, — что она находится в прямоугольной комнате, выложенной квадратной плиткой, и что где-то в этой комнате расположен торт. Всё что остаётся делать Челл, — слепо переходить с плитки на плитку в поисках торта.

Кроме того, так как действие происходит в Лаборатории Исследования Природы Порталов, на каждой стене комнаты расположен портал, связанный с противоположной стеной. Формально: рассмотрим комнату как прямоугольную сетку размера $W \times H$ и введем систему координат так, чтобы левая нижняя ячейка сетки имела координаты $(0, 0)$, а правая верхняя — $(W - 1, H - 1)$. За один шаг Челл может перейти в одну из четырёх соседних ячеек, причём попытка пройти сквозь стену приводит к тому, что Челл появляется с противоположной стороны комнаты. Например, шаг вниз из ячейки $(x, 0)$ приведёт в ячейку $(x, H - 1)$, а шаг влево из ячейки $(0, y)$ приведёт в ячейку $(W - 1, y)$.

Чтобы сделать испытание сложнее, GLaDOS не сообщила Челл ни размеры комнаты, ни координаты её изначального положения, ни координаты торта. Более того, так как у Челл завязаны глаза, а порталная технология достигла совершенства в своём развитии, Челл даже не может определить, прошла ли она очередным ходом через портал или нет.

Челл может найти торт, только оказавшись в одной клетке с ним. Помогите ей пройти последнее испытание!

Протокол взаимодействия

Это интерактивная задача.

Ваша программа будет общаться с тестирующей системой по протоколу, описанному ниже. Вам разрешается произвести **не более 200 000 ходов**.

Чтобы переместиться в соседнюю ячейку, выведите строку, содержащую ровно один символ, задающий направление перемещения: «U» — вверх; «D» — вниз; «L» — влево; «R» — вправо.

Затем вы должны считать строку, в которой будет находиться ровно один символ, обозначающий результат перемещения:

- «Y» — после произведённого хода Челл оказалась в клетке с тортом;
- «N» — после произведённого хода Челл оказалась в клетке, не содержащей торт;
- «E» — служебный символ, обозначающий, что после произведённого хода Челл всё ещё не нашла торт, а ваша программа превысила ограничение на количество ходов.

Обратите внимание, после считывания символа «Y» или «E» вы обязательно должны сразу завершить вашу программу. В противном случае, вердикт тестирующей системы может быть некорректным!

Гарантируется, что в клетке, в которой исходно находится Челл, нет торта

Система оценки

Обозначим за Q количество ходов, которое произвела ваша программа, прежде чем найти торт. Помимо общего ограничения $Q \leq 200\,000$, для каждой группы тестов может быть своё ограничение, связывающее Q и S , где $S = W \cdot H$ — площадь комнаты. При превышении любого из этих ограничений, ваша программа получает вердикт Wrong Answer.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Результаты
1	25	$10 \leq S \leq 20$	0	Первая ошибка
2	25	$10 \leq S \leq 300, \log_2 Q \leq 2 \log_2 S + 2$	0-1	Первая ошибка
3	25	$10 \leq S \leq 2000, \log_2 Q \leq 1.5 \log_2 S + 2.5$	0-2	Первая ошибка
4	25	$10 \leq S \leq 6000, \log_2 Q \leq \log_2 S + 5$	0-3	Первая ошибка

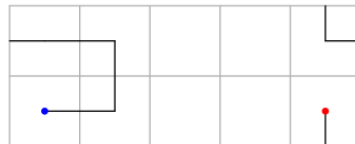
Обратите внимание, в ограничениях на Q во второй, третьей и четвёртой группе тестов фигурирует площадь комнаты в каждом конкретном тесте, а не её максимально возможное значение в данной группе тестов. Для установки правильности вашей программы жюри может запускать её произвольное количество раз для одной и той же комнаты, располагая в ней торт в различных местах.

Пример

стандартный ввод	стандартный вывод
N	R
N	U
N	L
N	L
Y	U

Замечание

Пояснение к примеру из условия. В примере из условия поле может выглядеть следующим образом:



Начальное положение Челл обозначено синей точкой, положение торта - красной.

Задача F. 1.5G — модем

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

Алиса должна передать Бобу целое неотрицательное число X , меньшее 10^{10} . Единственным способом связи у Алисы является старый 1.5G модем, который при передаче произвольным образом переставляет цифры в числе.

Алиса может отправить по модему число из не более, чем 2021 цифр (ведущие нули допустимы).

Боб получает от Алисы отправленное число с переставленными произвольным образом цифрами. Задача Боба — восстановить по полученному сообщению число X .

Формат входных данных

В первой строке входных данных задаётся одна строка — «Alice», если данные предназначены для Алисы, и «Bob», если данные предназначены для Боба.

В случае, когда данные предназначены для Алисы, вторая строка содержит целое неотрицательное число X , меньшее 10^{10} , заданное без ведущих нулей.

В случае, когда данные предназначены для Боба, вторая строка содержит строку s из цифр — отправленное Алисой число с переставленными цифрами. Строка не может быть пустой или состоять из более чем 2021 цифры.

Формат выходных данных

Если входные данные предназначены для Алисы, выведите непустую строку, состоящую из не более, чем 2021 цифр.

Если входные данные предназначены для Боба, выведите одно число — предполагаемое значение X .

Протокол взаимодействия

Ваше решение будет запущено два раза — в первый раз на данных, предназначенных для Алисы, во второй раз — на сгенерированных первым запуском данных, предназначенных для Боба. Передача каких-либо ещё данных между запусками запрещена.

Примеры

стандартный ввод	стандартный вывод
Alice 2021	66889
Bob 68986	2021