

Задача А. Разбиение массива

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан массив $A = [a_1, a_2, \dots, a_n]$, содержащий n натуральных чисел.

Требуется раскрасить элементы массива в два цвета таким образом, чтобы не существовало двух элементов x и y одного цвета, таких что x нацело делится на y и выполнялось равенство $\frac{x}{y} = p$, где p — простое число. Гарантируется, что такая раскраска существует.

Напомним, что целое число $p > 1$ называется простым, если оно имеет ровно два делителя: 1 и p .

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество элементов в массиве.
Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$) — элементы массива.

Формат выходных данных

Выведите описание разбиения массива на два множества в следующем формате.

Выведите n целых чисел, i -е из которых равняется 1, если элемент a_i надо раскрасить в первый цвет и 2, если элемент a_i надо раскрасить во второй цвет.

Если существует несколько подходящих раскрасок, вы можете вывести любую из них.

Система оценивания

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	9	$a_i \leq 2$ для всех i		первая ошибка
2	19	Гарантируется, что все a_i являются степенями некоторого простого числа p		первая ошибка
3	12	$a_i \leq 3$ для всех i	1	первая ошибка
4	13	$a_i \leq 4$ для всех i	1, 3	первая ошибка
5	21	$n \leq 10$		первая ошибка
6	26	нет	1–5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	2 1 1 2
1 20	1

Замечание

В первом примере есть два элемента первого цвета: 2 и 3, и два элемента второго цвета: 1 и 4. Элементы первого цвета не делятся нацело друг на друга. 4 нацело делится на 1, но их отношение не является простым числом.

Задача В. Последний рубеж

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.

Система оценивания

Подзадача	Баллы	Доп. ограничения	Оценка	Необх. подзадачи
0	0	Тесты из условия	потестовая	—
1	20	$Q = 100 \ 2 \leq N \leq 100$	подзадача	—
2	10	$Q = 100 \ 2 \leq N \leq 10^5$ В начале закрыты ровно два замка	подзадача	—
3	30	$Q = 90 \ 2 \leq N \leq 10^9$	подзадача	1, 2
4	20	$Q = 65 \ 2 \leq N \leq 10^{18}$	подзадача	1, 2, 3
5	20	$Q = 60 \ 2 \leq N \leq 10^{18}$	подзадача	1, 2, 3, 4

Для каждой подзадачи сообщаются набранные баллы, а также результат тестирования на первом непройденном тесте.

Примеры

стандартный ввод	стандартный вывод
3 0	? 2 ! 1
3 1	? 2 ! 2
5 0 0 1	? 2 ? ? ? ! 4

Задача С. Исправление одной ошибки

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Это задача с двойным запуском.

Ваше решение будет запущено два раза.

При первом запуске ему на вход будет передана строка x из нулей и единиц длиной не больше n . Программа должна вывести строку y из нулей и единиц, длина которой не более чем m . Число m не известно вашей программе и зависит от подзадачи, соответствующее значение указано в таблице системы оценивания. Если ваша программа выведет строку длиннее, чем m , она получит вердикт «Wrong answer».

Между запусками решения программа жюри внесет в строку y не более одной модификации, заменив ноль на единицу или единицу на ноль, получив, таким образом, строку z .

При втором запуске программе на вход будет подана строка z . Она должна восстановить исходную строку x и выдать её на выход.

Формат входных данных

При первом запуске на первой строке ввода находится число 1. На второй строке ввода находится строка x из нулей и единиц длины n ($10 \leq n \leq 100\,000$).

При втором запуске на первой строке ввода находится число 2. На второй строке ввода находится строка z из нулей и единиц длины не больше m ($10 \leq m \leq 300\,000$). Гарантируется, что эта строка равна строке y , выведенной программой при первом запуске, или получена из неё изменением ровно одного символа на противоположный.

Формат выходных данных

При первом запуске необходимо вывести строку y , которая позволит восстановить x после внесения в нее изменения. Длина строки y не должна превышать m .

При втором запуске по заданной строке z необходимо восстановить исходную строку x и вывести её.

Система оценивания

- 40 баллов. $n = 10$, $m = 30$
- 15 баллов. $n = 100\,000$, $m = 300\,000$
- 15 баллов. $n = 100\,000$, $m = 200\,000$
- 15 баллов. $n = 100\,000$, $m = 101\,000$
- 15 баллов. $n = 100\,000$, $m = 100\,017$

Примеры

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, если ваша программа выведет другую строку y , при втором запуске ввод также может быть другим.

Пример 1

Первый запуск.

стандартный ввод	стандартный вывод
1 0000000000	00000000000000000000000000000000

Второй запуск.

стандартный ввод	стандартный вывод
2 00000000000000000000000000000000	0000000000

Пример 2

Первый запуск.

стандартный ввод	стандартный вывод
1 0000000000	00000000000000000000000000000000

Второй запуск.

стандартный ввод	стандартный вывод
2 00000000000010000000000000000000	0000000000

Задача D. Угадай строку

Ограничение по времени: 15 секунд
Ограничение по памяти: 256 мегабайт

Эта задача с открытыми тестами.

Жюри загадало строку s . У этой строки вычислили хеш по следующей формуле: $h = (s[0] \cdot p^{n-1} + s[1] \cdot p^{n-2} + \dots + s[n-1] \cdot p^0) \pmod{q}$, где n — длина строки s . Вам даны значения h , p и q . Найдите такую строку s , что её хеш наиболее близок к значению h .

Формат входных данных

В единственной строке вам даны 3 целых числа h , p и q ($0 \leq h < p$, $1 \leq p \leq q \leq 10^{100}$).

Формат выходных данных

Ответ на каждый тест должен находиться в соответствующем файле: 01.txt, 02.txt, 03.txt, 04.txt, 05.txt, 06.txt, 07.txt, 08.txt, 09.txt, 10.txt.

Для того, чтобы послать ответ на некоторый набор тестов, выведете ответы в соответствующие файлы и упакуйте их в zip-архив. Обратите внимание, что архив не должен содержать иных файлов или папок, помимо ответов на тесты. Ответы должны лежать в корне архива. **То есть вы должны архивировать набор файлов, а не папку, их содержащую.** Также обратите внимание, что если у вас mac, то там в zip архив добавляется мусор, советуем на mac использовать приложение keka и там при архивации выбрать опцию убирания исходных файлов мака.

В единственной строке выведите строку s , состоящую из строчных букв английского алфавита, у которой хеш наиболее близок к h .

Система оценивания

В этой задаче 6 тестов, помимо примера из условия. За каждый тест ваш балл считается по следующей формуле:

Если ваш вывод не соответствует формату вывода, вы получите 0 баллов. Иначе, пусть хеш найденной вами строки равен t . Ваш балл за тест будет считаться по следующей формуле:

$$score = \left\lfloor 10 - \log_2 \left(1 + \frac{|t - h|}{q^{1/10}} \right) \right\rfloor$$

Пример

стандартный ввод	стандартный вывод
8 1 10	a

Замечание

Тест из примера имеет номер 0 и оценивается в 0 баллов. Если бы тест оценивался не нулем баллов, то хеш от строки a равен 7, балл считается по формуле $\left\lfloor 10 - \log_2 \left(1 + \frac{|7-8|}{10^{1/10}} \right) \right\rfloor = 9$.