

Задача А. Погоня в метро

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Это интерактивная задача. Параллельно с выполнением вашего решения жюри запускает проверяющую программу с которой вы обмениваетесь сообщениями через стандартный ввод и вывод. Подробнее о протоколе взаимодействия написано ниже. Также в конце условия вы можете посмотреть корректные примеры взаимодействия с проверяющей программой на разных языках программирования.

В прекрасной Метрополии будущего необходимость в машинистах, управляющих поездами метро, отпала. Благодаря развитию технологий, их заменил искусственный интеллект (ИИ). К сожалению, в один прекрасный день опасения писателей-фантастов сбылись — ИИ взбунтовался, и теперь где-то в метро ездит неуправляемый поезд. Страшно представить, чем это грозит городской транспортной системе! Ваша задача состоит в том, чтобы найти поезд в сложной системе метро и остановить неуправляемый ИИ.

В целях безопасности все остальные поезда были отправлены в депо, а все ветки, кроме той, на которой находится неконтролируемый поезд, были перекрыты, поэтому на данный момент метро Метрополии представляет из себя одну ветку (обычную прямую ветку без самопересечений) из n станций, последовательно пронумерованных от 1 до n , ровно на одной из которых находится поезд. Для поимки неуправляемого поезда вам требуется определить номер этой станции, после чего на путях будут установлены искусственные заграждения и поезд будет пойман.

Для определения нужной станции диспетчер Сара одолжила вам устройство, позволяющее вам выбрать произвольные числа l и r ($l \leq r$), после чего оно проверит, верно ли, что поезд находится на станции с номером между l и r . К сожалению, для перезарядки устройства требуется k минут (и вы используете его как только перезарядка завершается), поэтому между двумя применениями поезд может перебраться из той станции, где он сейчас находится, в любую станцию с номером отличающимся не более чем на k . Формально, если при некотором применении устройства поезд находился на станции x , то при следующем применении он может находиться на любой станции y , такой что $\max(1, x - k) \leq y \leq \min(n, x + k)$. При этом поезд не знает, что вы пытаетесь его поймать и совершает все перемещения согласно некоторому заранее составленному им плану.

В процессе изучения устройства вы выяснили, что оно было сделано очень давно, и сможет выдержать не более чем q использований, после чего оно сломается, а ваша задача будет считаться проваленной.

Сможете ли вы найти станцию, на которой находится поезд, за не более чем q использований устройства?

Протокол взаимодействия

При запуске решения на вход подаются три целых числа n ($1 \leq n \leq 10^{18}$), k ($0 \leq k \leq 10$) и q ($4500 \leq q \leq 15\,000$).

Чтобы использовать устройство, вы должны вывести через пробел два числа l и r ($1 \leq l \leq r \leq n$). В ответ на это вы получите либо строку «Yes», если между станциями с номерами l и r находится поезд, либо строку «No» иначе. Если $l = r$ и вы получили ответ «Yes», это значит, что вы точно определили станцию, на которой находится поезд, и ваша программа после этого должна немедленно завершиться.

Если ваша программа за q запросов не смогла точно определить станцию, на которой находится поезд, программа должна также немедленно завершиться, в противном случае вердикт тестирующей системы может быть любым.

После каждого запроса необходимо вывести перевод строки и сбросить 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. В точности соблюдайте формат взаимодействия с системой.

Пример

стандартный ввод	стандартный вывод
10 2 15000	
Yes	3 4
No	3 3
Yes	2 2

Замечание

В первом тесте поезд изначально находился на станции 3, после первого использования устройства переместился на станцию 2, а после второго — остался на месте.

Задача В. Кубики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Родители подарили Пете набор детских кубиков. Поскольку Петя скоро пойдет в школу, они купили ему кубики с буквами. На каждой из шести граней каждого кубика написана буква.

Теперь Петя хочет похвастаться перед старшей сестрой, что научился читать. Для этого он хочет сложить из кубиков её имя. Но это оказалось довольно сложно сделать — ведь разные буквы могут находиться на одном и том же кубике и тогда Петя не сможет использовать обе буквы в слове. Правда одна и та же буква может встречаться на разных кубиках. Помогите Пете!

Дан набор кубиков и имя сестры. Выясните, можно ли выложить её имя с помощью этих кубиков и если да, то в каком порядке следует выложить кубики.

Формат входных данных

В первой строке вводится число N ($1 \leq N \leq 100$) — количество кубиков в наборе у Пети. Во второй строке задано имя Петиней сестры — слово, состоящее только из больших латинских букв, не длиннее 100 символов. Следующие N строк содержат по 6 букв (только большие латинские буквы), которые написаны на соответствующем кубике.

Формат выходных данных

В первой строке выведите «YES» если выложить имя Петиней сестры данными кубиками можно, «NO» в противном случае.

В случае положительного ответа, во второй строке выведите M различных чисел из диапазона от 1 до N , где M — количество букв в имени Петиней сестры. i -е число должно быть номером кубика, который следует положить на i -е место при составлении имени Петиней сестры. Кубики нумеруются с 1, в том порядке, в котором они заданы во входных данных. Если решений несколько, выведите любое. Разделяйте числа пробелами.

Примеры

стандартный ввод	стандартный вывод
2 AB AAAAAB AAAAAA	YES 2 1
3 ANNY AAAAAA NNNNNN YYYYYY	NO

Задача С. Корень кубического уравнения

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Дано кубическое уравнение $ax^3 + bx^2 + cx + d = 0$ ($a \neq 0$). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

Формат входных данных

Во входном файле через пробел записаны четыре целых числа: $-1000 \leq a, b, c, d \leq 1000$.

Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 5 знаков после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
1 -3 3 -1	1.0000003749
-1 -6 -12 -7	-1.0000000111

Задача D. LCA-3

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

Подвешенное дерево — это ориентированный граф без циклов, в котором в каждую вершину, кроме одной, называемой *корнем* ориентированного дерева, входит одно ребро. В корень ориентированного дерева не входит ни одного ребра. *Отцом* вершины называется вершина, ребро из которой входит в данную.

(по материалам Wikipedia)

Дан набор подвешенных деревьев. Требуется выполнять следующие операции:

- 0 u v Для двух заданных вершин u и v выяснить, лежат ли они в одном дереве. Если это так, вывести вершину, являющуюся их наименьшим общим предком, иначе вывести 0.
- 1 u v Для корня u одного из деревьев и произвольной вершины v другого дерева добавить ребро (v, u) . В результате эти два дерева соединятся в одно.

Вам необходимо выполнять все операции online, т.е. вы сможете узнать следующий запрос только выполнив предыдущий.

Формат входных данных

На первой строке входного файла находится число n — суммарное количество вершин в рассматриваемых деревьях, $1 \leq n \leq 50000$. На следующей строке расположено n чисел — предок каждой вершины в начальной конфигурации, или 0, если соответствующая вершина является корнем. Затем следует число k — количество запросов к вашей программе, $1 \leq k \leq 100000$. Каждая из следующих строк содержит по три целых числа: вид запроса (0 — для поиска LCA или 1 — для добавления ребра) и два числа x, y . Вершины, участвующие в запросе можно вычислить по формуле: $u = (x - 1 + ans) \bmod n + 1$, $v = (y - 1 + ans) \bmod n + 1$, где ans - ответ на последний запрос типа 0 ($ans = 0$ для первого запроса).

Формат выходных данных

Для каждого запроса типа 0, выведите в выходной файл одно число на отдельной строке — ответ за этот запрос.

Пример

стандартный ввод	стандартный вывод
5	0
0 0 0 0 0	5
12	5
1 5 3	3
0 2 5	2
1 4 2	3
1 1 5	3
0 1 5	2
1 3 4	
0 1 5	
0 3 1	
0 4 2	
0 1 4	
0 5 2	
0 4 1	

Задача E. Невозрастающая подпоследовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (т.е. такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длиной $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделенные друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке выходного файла число k — длину максимальной невозрастающей подпоследовательности. Во второй строке должны быть выведены все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Пример

стандартный ввод	стандартный вывод
5	3
5 8 10 4 1	1 4 5

Задача F. Мега-инверсии

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Инверсией в перестановке p_1, p_2, \dots, p_N называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовём мега-инверсией в перестановке p_1, p_2, \dots, p_N тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Напишите алгоритм для быстрого подсчёта количества мега-инверсий в перестановке.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 100\,000$). Следующие N чисел описывают перестановку: p_1, p_2, \dots, p_N ($1 \leq p_i \leq N$), все p_i попарно различны. Числа разделяются переводами строк.

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке p_1, p_2, \dots, p_N .

Пример

стандартный ввод	стандартный вывод
4	4
4	
3	
2	
1	

Задача G. Игра пингвина

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Обратите внимание: эта задача интерактивная.

Пингвин Ксорий недавно придумал новую игру. У него есть n сосулек, занумерованных целыми числами от 1 до n . Каждая сосулька имеет температуру — целое число от 1 до 10^9 . **Ровно две** из этих сосулек особые: их температура равна y , в то время как температура всех остальных сосулек равна $x \neq y$. Вам нужно найти две особые сосульки. Вы можете не более **19** раз выбрать некоторое *непустое* подмножество сосулек и спросить у пингвина, чему равно побитовое исключающее ИЛИ (XOR) температур сосулек из этого подмножества.

Вам необходимо найти особые сосульки.

Формат входных данных

Первая строка содержит три целых числа n, x, y ($2 \leq n \leq 1000, 1 \leq x, y \leq 10^9, x \neq y$) — количество сосулек, температура обычных сосулек и температура особых сосулек.

Формат выходных данных

Чтобы сообщить пингвину свой ответ, выведите символ «!» (без кавычек), затем два целых числа p_1, p_2 ($p_1 < p_2$) — номера особых сосулек **в порядке возрастания**. Обратите внимание, что «!» и p_1 должны быть разделены пробелом; номера также должны быть разделены пробелом. После того, как вы сообщили ответ, ваша программа должна немедленно завершиться.

Протокол взаимодействия

Чтобы задать вопрос, выведите символ «?» (без кавычек), число c ($1 \leq c \leq n$), а затем c различных чисел p_1, p_2, \dots, p_c ($1 \leq p_i \leq n$) — номера сосулек, про которые вы хотите узнать информацию. Обратите внимание, что «?» и c должны быть разделены пробелом; номера также должны быть разделены пробелами.

После того, как вы задали вопрос, необходимо считать одно число — ответ на вопрос.

Обратите внимание, что вы можете задать не более **19** вопросов. В случае, если вы зададите больше 19 вопросов или зададите хотя бы один некорректный вопрос, ваше решение получит вердикт «Неправильный ответ».

Если в какой-то момент ваша программа считывает -1 как ответ, она должна немедленно завершиться (например, вызовом `exit(0)`). Вы получите вердикт «Неправильный ответ», и это будет означать, что вы задали больше 19 вопросов или задали некорректный вопрос. Если вы проигнорируете это, то можете получить любой вердикт, так как ваша программа продолжит читать из закрытого потока ввода.

Выше решение получит вердикт «Решение зависло», если вы не будете ничего выводить или забудете сделать операцию `flush` после вывода вопроса или ответа.

Чтобы выполнить операцию `flush`, можете использовать (сразу после вывода чисел и перевода строки):

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

Пример

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

Замечание

Для первого вопроса ответ равен $1 \oplus 2 \oplus 1 = 2$.

Для второго и третьего вопросов ответ равен 1, следовательно, особые сосульки имеют номера 1 и 3.

Задача Н. Транзисторы над Пекином

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме N этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа f транзистор обязательно разобьется при падении с этажа $f + 1$. Дополнительно поставлено условие, что разрешено проведение не более чем K испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

Формат входных данных

В первой и единственной строке входного файла содержатся два целых положительных числа N и K не превосходящих 2000.

Формат выходных данных

Выведите единственное число - ответ на поставленную задачу. Если для данных N и K возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите -1 .

Примеры

стандартный ввод	стандартный вывод
4 2	-1
4 3	2

Задача I. Сломанные роботы

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

У Кати есть n свечек, пронумерованных от 1 до n . Изначально они не горят.

Также у неё есть m роботов. i -й робот зажигает свечки $a_i, a_i + d_i, a_i + 2 \cdot d_i, \dots, a_i + t \cdot d_i$, где t — наибольшее целое число такое, что $a_i + t \cdot d_i \leq n$. Катя хочет приказать всем роботам сделать свою работу. Свечка горит, если хотя бы один из роботов её зажжёт.

К сожалению, ровно k роботов сломаны, но Катя не помнит, какие именно!

Найдите матожидание количества зажжённых свечек после того, как Катя прикажет всем роботам сделать свою работу, в предположении, что любой набор из k сломанных роботов равновероятен.

Это матожидание может быть представлено в виде p/q , где p и q — взаимно простые неотрицательные числа. Вы должны вывести $(p \cdot q^{-1}) \bmod (10^9 + 7)$, где q^{-1} — обратный к q по модулю.

Формат входных данных

Первая строка содержит три целых числа n, m, k ($1 \leq n, m \leq 2 \cdot 10^5, 1 \leq k \leq m$).

Следующие m строк содержат описание роботов. i -я из этих строк содержит два целых числа a_i и d_i ($1 \leq a_i, d_i \leq n$).

Формат выходных данных

Выведите одно число — ответ.

Пример

стандартный ввод	стандартный вывод
4 2 1 1 2 2 2	2

Замечание

В примере ровно один робот сломан.

Если сломан первый робот, то второй зажжёт свечки 2, 4.

Если сломан второй робот, первый зажжёт свечки 1, 3.

Таким образом, матожидание количества зажжённых свечек равно $\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 2 = \frac{2}{1}$ и $(2 \cdot 1^{-1}) \bmod (10^9 + 7)$ равно 2.