

Задача А. Задача Бахгольда

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

Задача Бахгольда формулируется очень просто. Дано целое положительное число n . Требуется представить его в виде суммы **максимального количества** простых слагаемых. Известно, что представление в виде суммы простых существует для всех целых положительных чисел, больших 1.

Напомним, целое положительное число k называется простым, если оно больше 1 и у него ровно два целых положительных делителя — 1 и k .

Формат входных данных

В единственной строке записано целое число n ($2 \leq n \leq 100\,000$).

Формат выходных данных

В первой строке выведите целое число k — максимально возможное количество простых слагаемых.

В следующей строке выведите k чисел — сами простые слагаемые. Их сумма должна быть равна n . Числа можно выводить в любом порядке. Если оптимальных решений несколько, разрешается вывести любое из них.

Примеры

стандартный ввод	стандартный вывод
5	2 2 3
6	3 2 2 2

Задача В. Строковая ловушка

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

Дана строка длины n . В начальный момент времени вы находитесь в позиции 1 (позиции нумеруются с единицы), на каждом шаге выполняется переход в другую позицию в соответствии с следующими правилами: если в строке есть еще одна или более позиций, буквы в которых совпадают с буквой в текущей позиции, то вы переходите случайную из них, иначе — двигаетесь на одну позицию вправо.

Можно ли выбраться из строки (под этим понимается, что вы в какой-то момент времени находитесь в позиции n после чего сдвигаетесь вправо) или же Вы попали в строковую ловушку, и вам придется блуждать по ней вечно?

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 10^5$) — длина строки.

Во второй строке дана строка s , строка состоит только из строчных букв латинского алфавита.

Формат выходных данных

Выведите «YES», если выбраться из строки возможно, и «NO» — в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 abc	YES
3 aaa	NO

Задача С. Параллелограмм возвращается

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

Давным-давно Лёша придумал задачу про параллелограмм. Входными данными в этой задаче являлись четыре целочисленные точки на плоскости, определяющие множество вершин некоторого параллелограмма ненулевой площади, не обязательно в порядке обхода.

У Лёши был очень хороший тест на эту задачу, но так получилось, что в файле с этим тестом была стёрта последняя строчка, так что вместо четырех точек осталось всего три. Но тест был настолько хороший, и он так сильно нужен Лёше, что он просит вас восстановить его.

Формат входных данных

Входные данные содержат три строки, в каждой из которых записаны два целых числа — координаты x_i и y_i ($-1000 \leq x_i, y_i \leq 1000$). Эти три точки не лежат на одной прямой, и никакие две из них не совпадают.

Формат выходных данных

В первой строке выведите число k — количество вариантов добавить к данным трём точкам четвёртую точку с целочисленными координатами так, чтобы получившееся множество точек определяло вершины некоторого параллелограмма ненулевой площади (точки не обязательно должны следовать в порядке обхода).

Далее выведите k строк, в каждой по два целых числа — координаты четвёртой точки.

Пример

стандартный ввод	стандартный вывод
0 0	3
1 0	1 -1
0 1	-1 1
	1 1

Замечание

Если вам требуется пояснение о том, что такое параллелограмм, посмотрите соответствующую статью в Википедии:

<https://ru.wikipedia.org/wiki/Параллелограмм>

Задача D. Рутинная задача

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

Могут ли олимпиадники заниматься настоящим программированием?

Определения:

Общий вид корректной сигнатуры метода на языке Java:

<тип возвращаемого значения> <название метода>(<тип аргумента №1> <имя аргумента №1>, <тип аргумента №2> <имя аргумента №2>, ..., <тип аргумента №n> <имя аргумента №n>);

<тип возвращаемого значения> ::= void | char | int | long | float | double | String | boolean | byte | short;

<тип аргумента> ::= char | int | long | float | double | String | boolean | byte | short;

<название аргумента> ::= a | ... | z;

<название метода> представляет из себя строку, состоящую из строчных латинских букв длины не больше 20 символов.

Гарантируется, что название метода не совпадает с названием ни одного из типов. Также гарантируется, что все аргументы имеют попарно различные имена.

Техническое задание:

Дана корректная сигнатура метода на языке Java. Требуется определить возможность с помощью одного выделения последовательного набора символов и его перетаскивания переместить один из аргументов метода на новую позицию так, чтобы сигнатура метода осталась корректной.

User story №1:

Задана сигнатура метода:

void method(int a, float b, char c);

Требуется переместить аргумент b на первую позицию. Для этого можно выделить подстроку:

void method(int a, float b, char c);

И переместить ее на первую позицию:

void method(float b, int a, char c);

User story №2:

Задана сигнатура метода:

int gcd(int a, int b);

Требуется переместить аргумент a на вторую позицию. Сделать это так, чтобы сигнатура метода осталась корректной, невозможно.

Формат входных данных

В первой строке задана корректная сигнатура метода на языке Java.

Во второй строке задано имя аргумента, который требуется переместить, и номер позиции, на которой он должна оказаться после перемещения. Гарантируется, что аргумент с таким именем присутствует в сигнатуре метода и что номер позиции не превосходит числа аргументов.

Формат выходных данных

В единственной строке выведите «YES» если возможно переместить аргумент на новую позицию с помощью описанной в техническом задании операции и выведите «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
<code>void method(int a, float b, char c); b 1</code>	YES
<code>int gcd(int a, int b); a 2</code>	NO

Задача Е. Рождественский подарок

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

Когда жизнь дает тебе лимоны, ты спрашиваешь "Зачем мне лимоны, где мое Рождество?!"

Это интерактивная задача. Здесь ваша программа должна по мере решения обмениваться информацией с программой жюри. Обратите внимание, что 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, «`flush(output)`» в Pascal, «`sys.stdout.flush()`» в Python.

Если вы еще не знали, НС обожает Рождество. Его бесконечная любовь к этому празднику способна преодолевать световые барьеры, искривлять течение времени и производить взрывы, разрушающие целые галактики.

Одна из самых важных вещей для НС в Рождество - это подарки. Ему стало известно, что Тамара и Малкольм спрятали его подарок где-то на заднем дворе. НС просто обязан проверить свой подарок до Рождества, ведь если подарок окажется не идеальным, то весь праздник будет испорчен! (Но еще есть время избить Тамару и Малкольма молотком и заставить их подарить НС на Рождество что-то действительно стоящее).

Задний двор представляет из себя прямоугольное поле, левый нижний конец которого находится в точке $(0, 0)$, а правый верхний в точке (n, m) , таким образом размер поля $(n+1) \times (m+1)$. К счастью, у НС есть Интерактивный Отыскатель Подарков — устройство, позволяющее определить расстояние от любой точки до подарка. Интерактивный Отыскатель работает следующим образом: НС называет ему точку (x, y) , а Отыскатель возвращает ему единственное число, равное $|x - X| + |y - Y|$, где (X, Y) - место нахождения подарка. До Рождества осталось мало времени, поэтому НС успеет сделать **не более чем 10 запросов**.

НС очень взволнован что может не успеть найти подарок до Рождества, поэтому он просит вас помочь ему!

Формат входных данных

В первой строке содержатся два целых числа n и m ($0 \leq n \leq 10^9, 0 \leq m \leq 10^9$).

После каждого запроса вашей программы программа жюри выводит единственное число, равное расстоянию от запрошенной вами точки до подарка как описано в условии задачи.

Формат выходных данных

Каждый запрос должен состоять из знака вопроса (?), после которого через пробел необходимо вывести ровно два целых числа x и y ($0 \leq x \leq n, 0 \leq y \leq m$) — точка, расстояние от которой до подарка вы хотите узнать.

Если вы уверены что нашли точку, в которой находится подарок, вы должны вывести ответ и завершить работу программы. Вывод ответа должен начинаться с восклицательного знака (!), после которого через пробел должны быть выведены два целых числа X и Y ($0 \leq X \leq n, 0 \leq Y \leq m$).

Вывод ответа учитывается в общем числе запросов.

Если ваш запрос не будет удовлетворять описанным ограничениям, то Интерактивный Отыскатель Подарков сломается, вы не найдете подарок, Рождество будет испорчено, а ваша программа получит вердикт Wrong Answer.

Напоминаем, что после каждого запроса и вывода ответ следует очищать потоковый буфер.

Примеры

стандартный ввод	стандартный вывод
2 2 ответ системы: 1 ответ системы: 3 ответ системы: 0	запрос участника: ? 1 1 запрос участника: ? 2 0 запрос участника: ? 1 2 запрос участника: ! 1 2
1 2 ответ системы: 1 ответ системы: 1 ответ системы: 2	запрос участника: ? 1 1 запрос участника: ? 0 2 запрос участника: ? 0 1 запрос участника: ! 1 2

Замечание

Строки «запрос участника» и «ответ системы» в примере приведены лишь для того, чтобы было более понятно, в каком порядке выводятся сообщения. При решении задачи вам не нужно выводить эти строки, программа жюри также не будет выводить такие строки.

Задача F. Снятся ли богам торты?

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

Можете представить, насколько сложно сделать торт? Нет, еще сложнее, чем вы представили.

А вот главный королевский кондитер точно представлял. Он готовил торты с детства, самых разных форм, цветов и вкусов. Когда его спросили, как у него получается делать такие сложные торты, он просто ответил: "Я верю, что Ника, богиня победы, очень любит торты - поэтому из кухни я всегда выхожу победителем".

Но однажды ему заказали особо необычный торт. Этот торт должны были повесить в комнате королевского первенца, поэтому он должен был быть выполнен в виде плоской картины из $N \cdot M$ кубиков попарно различных цветов, каждый кубик имел свой неповторимый цвет.

Кондитер разработал специальный аппарат для выдавливания кубиков нужной формы. Для упрощения конструкции материалы разных цветов вводились в аппарат заранее в неизменном после ввода порядке.

В тот знаменательный день кондитер задал порядок цветов в аппарате и отправился создавать сей шедевр. Он рассчитывал, что торт будет готовиться горизонтально и он сможет сам определять порядок заполнения торта, поэтому цвета в аппарат он задал в случайном порядке.

Но оказалось, что рамка была уже жестко закреплена на стене, поэтому единственным способом сделать торт было выдавливать кубики один за одним сверху, после чего кубик летел вниз и останавливался, как только достигал нижней границы рамки или верхней грани другого кубика. Кондитер мог выбрать колонку, куда он сбросит очередной кубик, но повлиять на порядок кубиков в колонке не мог никак.

Это был провал! Не каждый ведь порядок цветов может привести к созданию картины по заданному эскизу. Но отступать некуда - весь королевский двор собрался посмотреть, как кондитер приготовит свое самое великое творение.

Времени до начала не так и много, поэтому он просит вас поскорее определить по заданной картине и порядку цветов в аппарате, выйдет ли он победителем и в этот раз или его спасет только божественное вмешательство?

На всякий случай приведем формальное описание алгоритма получения картины для незнакомых с тонкостями кондитерского искусства:

- Цвета в аппарате заданы в определенном порядке, цвет C_k обрабатывается только после обработки всех цветов $C_1, C_2, \dots, C_{k-1}$;
- Выбирается колонка j ($1 \leq j \leq M$), куда будет выдавлен кубик цвета C_k ;
- Кубик занимает позицию в строке N , если до этого колонка была пустой, и в строке i , если позиции в данной колонке в строках $N, N-1, \dots, i+1$ заняты обработанными ранее кубиками.

Формат входных данных

В первой строке заданы два числа N и M ($1 \leq N \cdot M \leq 10^5$).

Следующие N строк содержат по M чисел A_{ij} ($1 \leq A_{ij} \leq N \cdot M$) - цвета кубиков в соответствующем ряду картины.

В последней строке задано $N \cdot M$ чисел C_k ($1 \leq C_k \leq N \cdot M$) - порядок цветов в аппарате кондитера, в котором он будет выдавливать кубики для картины.

Гарантируется, что все A_{ij} и C_k различны.

Формат выходных данных

Выведите в единственной строке «YES» (без кавычек), если можно получить картину с заданными эскизом и порядком цветов, и «NO» (без кавычек) в ином случае.

Примеры

стандартный ввод	стандартный вывод
2 2 1 2 3 4 3 4 1 2	YES
2 2 1 2 3 4 3 1 4 2	YES
2 3 1 2 3 4 5 6 6 1 5 2 4 3	NO

Задача G. Порядок задач

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

Имеется n задач по программированию, обозначенных числами от 1 до n , где 1 — самая простая задача, а n — самая сложная. Какой же должен быть порядок этих задач в контексте?

Егор считает, что оптимальный порядок задач выражается перестановкой $\{a_i\}$, а Денис настаивает на том, что он должен быть равен $\{b_i\}$. Если в итоге будет выбран порядок задач, выражающийся перестановкой $\{p_i\}$, то ненависть Егора к контексту будет равна $\sum_i |p_i - a_i|$, а ненависть Дениса — $\sum_i |p_i - b_i|$.

Окончательное решение почему-то осталось за Еленой, и она намерена минимизировать суммарную ненависть Егора и Дениса. Какой порядок задач она должна выбрать?

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 100000$) — длина перестановок.

Во второй строке записаны n чисел a_i — оптимальный порядок задач для Егора.

В третьей строке записаны n чисел b_i — оптимальный порядок задач для Дениса.

Формат выходных данных

Выведите n различных целых чисел от 1 до n через пробел — перестановку $\{p_i\}$, которая минимизирует суммарную ненависть.

Пример

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

Задача Н. Расширенный парадокс Монти-Холла

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

Парадокс Монти Холла — одна из известных задач теории вероятностей, решение которой, на первый взгляд, противоречит здравому смыслу.

Представьте, что вы стали участником игры, в которой вам нужно выбрать одну из трёх дверей. За одной из дверей находится автомобиль, за двумя другими дверями — козы. Вы выбираете одну из дверей, например, номер 1, после этого ведущий, который знает, где находится автомобиль, а где — козы, открывает одну из оставшихся дверей, например, номер 3, за которой находится коза. После этого он спрашивает вас — не желаете ли вы изменить свой выбор и выбрать дверь номер 2? Увеличатся ли ваши шансы выиграть автомобиль, если вы примете предложение ведущего и измените свой выбор?

С одной стороны, после того, как ведущий открыл дверь, осталось две закрытых двери - и вы можете ошибочно предположить, что вероятность автомобиля за каждой — $\frac{1}{2}$. Объяснение простое — изначально вероятность нахождения автомобиля за каждой из дверей — $\frac{1}{3}$, и она осталась такой после того, как ведущий открыл одну из оставшихся дверей. Соответственно при таком раскладе автомобиль находится за выбранной вами дверью с вероятностью $\frac{1}{3}$ и за оставшейся с вероятностью $\frac{2}{3}$.

Приведем еще более наглядный пример: у вас есть тысяча дверей, вам разрешено выбрать одну, после чего ведущий открывает 998 из 999 оставшихся дверей. Вероятность того, что автомобиль за вашей дверью — $\frac{1}{1000}$. Соответственно, вероятность, что автомобиль находится за одной из оставшихся 999 равна $\frac{999}{1000}$. А т.к. из оставшихся ведущий нам оставил лишь одну дверь, то именно на нее приходится эта вероятность.

Теперь перейдем к нашей задаче, перед вами n дверей - вам разрешено открыть k из них, после этого ведущий обязательно откроет m ($m < n - k$) из оставшихся дверей, после чего предложит поменять любое количество выбранных дверей. Какая вероятность вашего выигрыша при оптимальной стратегии?

Формат входных данных

В единственной строке даны три числа n, k, m ($1 \leq n, k, m \leq 10^5, k + m < n$).

Формат выходных данных

В единственной строке выведите вероятность своей победы при оптимальном выборе. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 1 1	0.6666666666666667
1000 1 998	0.999
5000 2345 1234	0.7158

Задача I. Сколько весит карма?

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

Ур и Ар были странствующими магами, неразлучными друзьями с самого раннего детства. Они приходили в какой-нибудь городок, показывали там всякие фокусы, танцы с бубнами, проводили сеансы телепатии - в общем, по всякому развлекали народ, а взамен получали кров и пищу.

Но в одном городке все пошло не по плану. Оказалось, что горожане поклонялись могучей богине справедливости и правосудия Анасте, и считали любую магию несправедливостью по отношению к суровой реальности. Ур и Ар были схвачены, приведены в храм Анасты и оставлены дожидаться пришествия богини.

И ведь дождались! Ровно в полночь Анаста явилась к напуганным до смерти магам. Она была слепа (как и всякое правосудие), а в руке держала невероятно большие рычажные весы. Когда человек вставал на одну чашу весов, на другой появлялись совершенные им несправедливости.

Ур и Ар не были злодеями, убивавшими детей и грабившими беззащитных стариков, но и в их жизни были мелкие косяки со справедливостью. Косяки никогда не повторялись, ведь друзья всегда хотели попробовать что-нибудь новое. Некоторые вещи они совершали вместе, а что-то Ур (или Ар) делал за двоих, беря на себя двойную ответственность.

Поэтому, когда Ур и Ар встали на разные чаши весов, на чашах появилось ровно $2N$ несправедливостей (по N на каждой чаше), занумерованных от 1 до n (каждая несправедливость встречается ровно 2 раза).

Анаста заявила, что когда весы придут в равновесие, она решит их судьбу. Ур и Ар подумали, что пока весы колеблются, они могут незаметно сделать набор несправедливостей и даже порядок расположения несправедливостей в наборах на чашах полностью одинаковым. Тогда Анасте придется выбрать им полностью одинаковую судьбу, и друзья смогут и дальше продолжить свои приключения вместе.

Весы оказались довольно неустойчивы, поэтому за одно действие Ур может перекинуть только одну несправедливость Ару, а Ар одновременно перекинуть несправедливость со своей чаши Уру.

На кону судьба их дружбы, поэтому Ур и Ар просят помочь вас и подсказать им возможный порядок перекидывания несправедливостей между чашами. Помните, что весы Анасты скоро придут в равновесие, поэтому маги успеют сделать не более $4n$ перекидываний.

Формат входных данных

В первой строке вводится число n ($1 \leq n \leq 10^5$) - количество несправедливостей на каждой из чаш весов.

В следующей строке вводятся n чисел a_i ($1 \leq a_i \leq n$) - порядок номеров несправедливостей на чаше весов с Уром.

В последней строке вводятся n чисел b_i ($1 \leq b_i \leq n$) - порядок номеров несправедливостей на чаше весов с Аром.

Гарантируется, что среди чисел a_i и b_i есть ровно два числа 1, два числа 2, ..., два числа n .

Формат выходных данных

В первой строке выведите k ($0 \leq k \leq 4n$)- количество действий магов, которое приведет к совпадению итоговых наборов несправедливостей и порядка расположения несправедливостей в наборах.

В каждой из следующих k строк выведите по два числа f_i и s_i - позиции несправедливостей на чаше Ура и Ара соответственно, которые необходимо поменять местами в i -ом действии.

Пример

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

Замечание

Рассмотрим пример:

Перед первым действием $a = [1, 1, 2]$, $b = [2, 3, 3]$.

1) $f_1 = 2$, $s_1 = 3$ - меняются местами a_2 и b_3 - $a = [1, 3, 2]$, $b = [2, 3, 1]$;

2) $f_2 = 3$, $s_2 = 3$ - меняются местами a_3 и b_3 - $a = [1, 3, 1]$, $b = [2, 3, 2]$;

3) $f_3 = 3$, $s_3 = 1$ - меняются местами a_3 и b_1 - $a = [1, 3, 2]$, $b = [1, 3, 2]$;

4) $f_4 = 3$, $s_4 = 3$ - меняются местами a_3 и b_3 - $a = [1, 3, 2]$, $b = [1, 3, 2]$.

Обратите внимание, что маги имели право остановиться уже после 3-го действия, так как наборы уже полностью совпадали.

Задача J. Голосование

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

В организации Alternative Cake Manufacturing (АСМ) работает n сотрудников. Сейчас там проводится очень важное голосование, и все ведущие мировые СМИ пытаются спрогнозировать его результат.

Так получилось, что каждый из сотрудников имеет ярко выраженные политические взгляды и принадлежит либо к фракции депубликанцев, либо к фракции ремократов, и каждой фракции выгоден свой результат голосования. Процедура голосования устроена крайне необычно:

1. Каждый из n сотрудников высказывается по очереди, начиная с сотрудника номер 1 и заканчивая сотрудником номер n . При этом если к моменту, когда сотруднику пора высказаться, его уже лишили права голоса, то он просто пропускает свой ход (и уже никак не участвует в голосовании в последствии).
2. Высказывание сотрудника состоит в том, чтобы лишить права голоса любого другого сотрудника или ничего не сделать. Лишать права голоса можно как тех кто уже высказался, так и тех, кому только предстоит высказаться. Лишение права голоса действует в течение всего процесса голосования.
3. Когда все сотрудники выскажутся, процедура повторяется: снова по очереди высказываются все от сотрудника 1 до сотрудника n , кто ещё не лишился права голоса.
4. Повторения процедуры происходят до тех пор, пока не останется только один сотрудник с правом голоса, то он в одиночку решает исход голосования. Разумеется, он принимает решение в пользу своей фракции.

Зная порядок голосования и то, что все сотрудники действуют оптимально (и знают порядок голосования и кто принадлежит какой фракции), вы должны предсказать, какая фракция одержит верх.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 200\,000$) — количество сотрудников.

Далее записана строка из n символов, i -й из которых равен «D», если i -й человек поддерживает депубликанцев, либо «R», если i -й человек поддерживает ремократов.

Формат выходных данных

Выведите «D», если на голосовании будет принято решение, устраивающее фракцию депубликанцев, и «R», если ремократов.

Примеры

стандартный ввод	стандартный вывод
5 DDRRR	D
6 DDRRRR	R

Замечание

Рассмотрим один из вариантов голосования:

1. 1-й сотрудник лишает голоса сотрудника 5.
2. 2-й сотрудник лишает голоса сотрудника 3.

3. 3-й сотрудник пропускает ход, так как его лишил голоса сотрудник 2.
4. 4-й сотрудник лишает голоса сотрудника 2.
5. 5-й сотрудник пропускает ход, так как его лишил голоса сотрудник 1.
6. 1-й сотрудник опять высказывается и лишает голоса сотрудника 4.
7. Только сотрудник 1 остался с правом голоса, поэтому голосование заканчивается принятием решения в пользу фракции депубликанцев.

Задача К. Договор со смертью

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

Смерть — неумолимая сила, с которой невозможно договориться... или возможно?

Когда Смерть пришла в дом к великому математику своей эпохи Кирюну д'Элю, он был готов к встрече с ней. Он прожил долгую жизнь, полную свершений и открытий, его имя было известно во всем мире.

Увидев гостя, Кирюн сказал, что вот-вот сделает новое открытие, которое навсегда победит квадрирование мыслей. Все, что осталось — запустить программу и узнать, какая цифра стоит в конце десятичной записи дроби A/B в системе счисления C . Кирюн утверждал, что это займет всего несколько секунд, и очень просил Смерть дать ему возможность дожидаться окончания вычислений, он даже обещал включить ее в соавторы этого поистине величайшего открытия всех времен и народов.

Вообще Смерти было даже приятно получить такое предложение — быть соавтором открытия с самим Кирюном — поэтому она согласилась. Ведь что значат несколько секунд перед вечностью, которая будет за ними. Но чем дольше программа считала результат, тем больше у Смерти закрадывались подозрения, что Кирюн обманул ее и эта дробь в заданной системе счисления имеет бесконечную десятичную запись.

Смерть не особо сильна в математике, но и спорить с великим Кирюном она пока не хочет, поэтому просит вас поскорее ответить, не превратятся ли эти несколько секунд ожидания в вечность.

Формат входных данных

В единственной строке вводятся три целых числа A , B , C ($1 \leq A, B \leq 10^9, 2 \leq C \leq 10^9$) — числитель, знаменатель дроби и основание системы счисления соответственно.

Формат выходных данных

Выведите «YES» (без кавычек), если десятичная запись дроби A/B является конечной в системе счисления C , и «NO» (без кавычек) в ином случае.

Примеры

стандартный ввод	стандартный вывод
1 2 10	YES
5 9 10	NO
5 9 3	YES

Задача L. Выбор столицы

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

Страна R состоит из n городов и $n - 1$ двусторонних дорог, расположенных так, что из любого города можно добраться до любого другого. Президент этой страны решил выбрать столицу. Как только столица будет выбрана, все жители страны съедутся на парад в столицу. Известно, что в i -м городе проживает v_i жителей. Решение, какой же именно город станет столицей, еще не принято, так как еще не до конца понятно, что же для страны будет выгоднее: с одной стороны, можно неплохо простимулировать транспортную систему страны, если сделать столицу в достаточно удаленном и малонаселенном городе, а с другой стороны, столица, расположенная в крупном городе поближе к центру, будет быстро развиваться и тоже потянет экономику вверх. Поэтому президент хочет посчитать для каждого города i , сколько суммарно дорог преодолеют все жители страны, чтоб добраться до столицы, если сделать ее в городе i . Без сомнения, обладая этой информацией, он примет правильное решение и приведет страну к успеху и процветанию.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 100000$) — количество городов в стране.

В следующей строке записаны n чисел v_i ($1 \leq v_i \leq 10^8$) — количества жителей в каждом городе.

В каждой из следующих $n - 1$ строк записаны два числа a_i и b_i ($1 \leq a_i, b_i \leq n$) — номера городов соединенных дорогой.

Формат выходных данных

Выведите n чисел. i -е число должно равняться суммарно пройденному расстоянию всех жителей страны до города i .

Пример

стандартный ввод	стандартный вывод
5 7 3 2 5 4 1 2 1 3 3 4 3 5	23 38 22 33 35

Задача М. Побег с аукциона

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

Сегодня проходит аукцион, в котором принимают участие n человек. Аукцион проходит по классическим правилам. Известно, что было сделано n ставок, при этом не гарантируется, что каждая ставка была сделана уникальным участником, то есть какие-то участники могли и вовсе не делать ставок.

Каждая ставка определяется парой целых чисел (a_i, b_i) , где a_i — номер участника, сделавшего ставку, а b_i — величина ставки. Все ставки даны в хронологическом порядке, то есть $b_i < b_{i+1}$ для всех $i < n$. Кроме того, ни один из участников не делал две ставки подряд (не перебивал свою ставку), то есть $a_i \neq a_{i+1}$ для всех $i < n$.

Из праздного любопытства вы заинтересовались следующим вопросом: кто и с какой ставкой выиграет в аукционе, если бы некоторые из участников на самом деле отсутствовали? При этом считается, что этих участников и их ставок просто не было, а среди оставшихся участников новых ставок не появится.

Обратите внимание, что если после мысленного вычёркивания каких-либо участников получится ситуация, когда кто-то перебивает свою ставку, то необходимо считать, что этого не происходит и из серии последовательных ставок одного и того же человека останется только первая. Для лучшего понимания, посмотрите примеры.

У вас есть несколько подобных предположений, вычислите ответ для каждого из них.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 200\,000$) — количество участников аукциона и сделанных ставок.

В следующих n строках расположены по два числа a_i и b_i ($1 \leq a_i \leq n, 1 \leq b_i \leq 10^9, b_i < b_{i+1}$) — номер участника, сделавшего i -ю ставку, и размер этой ставки.

В следующей строке дано число q ($1 \leq q \leq 200\,000$) — количество запросов.

В каждой из следующих q строк дано число k ($1 \leq k \leq n$), а затем k чисел l_j ($1 \leq l_j \leq n$) — количество отсутствующих участников, и их номера. Все номера в пределах одного запроса различны.

Гарантируется, что сумма всех k во всех запросах не превосходит 200 000.

Формат выходных данных

Для каждого запроса выведите два числа в одной строке через пробел — номер победителя и величину его ставки. Если победитель отсутствует (то есть не было сделано ни одной ставки), выведите два нуля.

Примеры

стандартный ввод	стандартный вывод
6 1 10 2 100 3 1000 1 10000 2 100000 3 1000000 3 1 3 2 2 3 2 1 2	2 100000 1 10 3 1000
3 1 10 2 100 1 1000 2 2 1 2 2 2 3	0 0 1 10

Замечание

Пояснение к первому примеру:

- В первом запросе отсутствует только участник с номером 3, тогда последовательность ставок будет выглядеть следующим образом:

- 1 10
- 2 2 100
3. 1 10 000
4. 2 100 000

В такой ситуации побеждает участник с номером 2, при этом его ставка составляет 100 000.

- Во втором запросе отсутствуют участники 2 и 3, тогда последовательность ставок будет выглядеть следующим образом:

1. 1 10
2. 1 10 000

Здесь победил участник с номером 1, но поскольку участнику не выгодно перебивать свою ставку, то выигрышной ставкой является 10, а не 10 000.

- В третьем запросе отсутствуют участники 1 и 2, тогда последовательность ставок будет выглядеть следующим образом:

1. 3 1 000
2. 3 1 000 000

Побеждает участник 3 со ставкой 1 000.