Задача A. Сотовые вышки

Имя входного файла:	`cell.in`
Имя выходного файла:	`cell.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Часто когда мы выбираемся на природу и отдых, мы стараемся избавиться от цивилизации, выключив телефоны и планшеты с безлимитными интернетом. Однако есть некоторые люди, которые не могут просто взять и отключить свой телефон, им просто обязательно нужно следить за новостями в соцсетях, постить сообщения и читать сообщения своих друзей.

Как-то раз Денис со своими друзьями собрался посетить заповедник им. Котова Тимофея Петровича, который находится в S километрах от города. Среди друзей Дениса есть Никита, который и минуты не может прожить без достижений цивилизации. Перед поездкой он решил подготовиться и посмотреть, в каких местах на пути до заповедника располагаются вышки. Таким образом Никита получил N чисел — расстояния от начала поездки до каждой вышки. На удивление дорога оказалась прямой, поэтому во время поездки мобильные устройства будут подключаться к самой близкой вышке. Никиту заинтересовал вопрос, на каком расстояния от точки отправления будет находиться наихудший сигнал.

Формат входного файла

В первой строке входного файла содержится два положительных целых числа N, S — количество сотовых вышек и расстояние от точки отправления до заповедника ($N, S \leq 299999$). В следующей строке содержится N целых неотрицательных чисел в порядке возрастания — расстояния от точки отправления в заповедник до очередной вышки. Расстояния неотрицательны и не превышают S.

Формат выходного файла

В выходной файл выведите единственное число с двумя знаками после запятой — расстояние от точки старта, на котором будет наблюдаться наихудший сигнал. Если существует несколько вариантов ответа, выведите тот, у которого расстояние от точки старта будет наименьшим.

Примеры

cell.in	cell.out
3 4 0 2 4	1.00

Задача B. Облако

Имя входного файла:	`cloud.in`
Имя выходного файла:	`cloud.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Константин Константинович Константинопольский — ученый-метеоролог. Недавно он увлекся строительством летательных аппаратов и построил радиоуправляемый зонд для исследования кучевых облаков.

Для исследования облака Константин Константинович поднимает аппарат в воздух, направляет его полёт параллельно поверхности Земли (поскольку высота небольшая, Землю можно считать плоской), после чего ориентирует аппарат таким образом, чтобы он пролетел сквозь облако по прямой линии. Константин Константинович понимает, что пока аппарат находится внутри облака, менять его курс рискованно, поскольку облако может исказить управляющие сигналы с пульта.

Так как качество исследования облака напрямую зависит от количества собранных данных, Константин Константинович хочет узнать, сколько времени аппарат проведёт в полёте внутри облака.

Для упрощения задачи Константин Константинович считает поверхность Земли плоской, а облако представляет в виде многоугольника на плоскости (необязательно выпуклого). Поскольку аппарат сравнительно мал, Константин Константинович считает его материальной точкой с некоторым заданным изначально направлением движения. Вам требуется по координатам точек многоугольника-облака, начальным координатам аппарата и вектору его скорости определить, какой путь аппарат проведёт внутри облака. При этом путь по границе облака учитывать не нужно.

Формат входного файла

В первой строке входного файла задано одно число N — количество вершин в многоугольнике, описывающем облако ($1 \le N \le 100$). Далее в N строках расположено по два числа x_i, y_i — координаты i-той вершины многоугольника (в порядке обхода). После этого в одной строке задается четыре числа x_0, y_0, v_x, v_y — координаты начального положения аппарата и его вектора скорости. Все координаты являются вещественными числами, не более чем с тремя знаками после десятичной точки. Все координаты не превосходят 1000 по модулю.

Формат выходного файла

Выведите единственное число — длину пути, которую аппарат проведёт внутри облака. Ваш ответ будет считаться корректным, если абсолютная или относительная погрешность ответа будет не более 10^{-5}.

Примеры

cloud.in	cloud.out
4 0 0 0 1 1 1 1 0 -1 -1 1 1	1.4142135624
4 0 0 0 1 1 1 1 0 -1 0 1 0	0.00000
5 0 0 1 10 2 1 3 10 4 0 -1 1 1 0	3.8000000000

Примечание

В примере 2 иллюстрируется условие, что путь по границе облака учитывать не нужно.

В примере 3 демонстрируется, что путь аппарата в облаке может состоять из нескольких отрезков.

Задача C. Гараж в снегу

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У вовиного дедушки есть гараж в кооперативе, в котором он держит свою машину. За зиму выпало очень много снега и дедушка попросил Вову почистить въезд в гараж от снега. Вова вооружился лопатой и бодро отправился к гаражам. По прибытии на место Вова обнаружил, что абсолютно все гаражи завалены доверху снегом и понять, какой же из них дедушкин, решительно невозможно. Всего Вова обнаружил N гаражей, расположенных в ряд вдоль дороги.

Вова вспомнил, что у гаражей определенно были номера. Позвонив дедушке, Вова узнал, что дедушкин гараж имеет номер X, а сами гаражи нумеруются в возрастающем порядке (но необязательно подряд), начиная с некоторого гаража i. Например, номера гаражей могут выглядеть вот так: `15 20 23 2 5 7 11` (в данном случае семь гаражей, нумерация начинается с четвёртого).

Откопать номер, написанный на гараже, намного проще, чем откопать гараж целиком, потому Вова решил сначала найти дедушкин гараж, а потом уже его откапывать. Помогите ему сделать это.

Формат входного файла

Это интерактивная задача, ваше решение будет взаимодействовать со специальной программой, подготовленной жюри. После запуска на стандартный поток ввода вы получите два числа N и X — общее количество гаражей в кооперативе и номер гаража вовиного дедушки ($1 \le N \le 10^6; 1 \le X \le 10^9$).

После этого вы можете отправлять Вову откапывать номер гаража, расположенного на месте i ($1 \le i \le N$). Для этого выведите число i, а за ним — перевод строки. В ответ на этот запрос вам на стандартный поток ввода придёт ответ a_i — номер гаража, расположенного на месте i ($1 \le a_i \le 10^9$). Помните, что гаражи нумеруются по возрастанию, но не подряд, а нумерация начинается не с первого гаража в кооперативе. Номера гаражей — целые положительные числа. Гарантируется, что гараж с номером X в кооперативе есть. Номера гаражей различны.

После того, как вы в ответ получите число X (искомый номер гаража), ваша программа должна завершиться. Если ваша программа завершится до того, как гараж будет найден, тест будет не пройден. Также, вы можете использовать не более 30 запросов (после этого Вова устанет и не сможет откопать дедушкин гараж).

Не забывайте сбрасывать буфер вывода после вывода каждого запроса. Это делается при помощи функции `fflush(stdout);` в C++ и `flush(output);` в Pascal.

Примеры

stdin	stdout
7 20 2 15 20	4 1 2

Примечание

В примере приведен пример диалога вашей программы и программы жюри (гаражи занумерованы так: `15 20 23 2 5 7 11`). Первым запросом программа участника отправляет Вову откапывать номер четвёртого гаража и получает в ответ номер 2. После второго запроса выясняется номер первого гаража 15, а после третьего запроса искомый гараж оказывается найден.

Задача D. НОД

Имя входного файла:	`gcd.in`
Имя выходного файла:	`gcd.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Анна Сергеевна — преподаватель математики средней школы в одном провинциальном мегаполисе. Следующее занятие в пятом классе будет посвящено нахождению наибольшего общего делителя.

Анна хочет дать детям задание на вычисление наибольшего общего делителя некоторого набора чисел. Для этого она выписала набор n чисел, но потом осознала, что n чисел — это слишком много для пятиклассника, поэтому она хочет удалить одно число из набора так, чтобы наибольший общий делитель оставшихся чисел был бы максимально возможным.

Анна Сергеевна и сама бы справилась с этой простой задачей, но очень устала, поэтому она просит вас помочь ей.

Формат входного файла

В первой строке входного файла находится натуральное число n — изначальное количество элементов в наборе, ($2 \le n \le 100000$). Во второй строке находится n чисел $a_1, a_2, \ldots, a_n$ — изначальный набор a, ($1 \le a_i \le 10^9$).

Формат выходного файла

В единственной строке выходного файла выведите два числа — индекс удаляемого элемента из набора и наибольший общий делитель оставшегося набора чисел. Если ответов несколько, то выведите тот, где индекс удаляемого элемента наименьший.

Примеры

gcd.in	gcd.out
4 4 1 4 8	2 4

Примечание

В примере после удаления числа 1 останется набор `4 4 8`, наибольший общий делитель в котором равен 4.

Задача E. Последовательность доктора Монта

Имя входного файла:	`sequence.in`
Имя выходного файла:	`sequence.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Михаил Афанасьевич Монт — ведущий специалист в области комбинаторных последовательностей, доктор физико-математических наук

Доктор Монт совершил научный прорыв в области комбинаторных последовательностей! Открытая им последовательность поможет приблизиться к решению задачи об оптимальной нумерации произвольных объектов, более известной как задача «О n слонах». Благодаря этому открытию Михаил Афанасьевич, возможно, получит Елефантовскую премию — самую престижную премию в области комбинаторных последовательностей.

Однако, М. А. Монту нужна ваша помощь — для более детального изучения последовательности ему нужно получить n–ый элемент этой последовательности, при этом n может быть очень велико!

Как? Вы не знаете формулу этой последовательности? Вы что, с Земли упали? Ладно, уговорили, мы напомним вам её определение.

Последовательность a называется последовательностью *М. А. Монта*, если выполняется следующее:

1. $a_1 = 1$;
2. Последовательность a — строго возрастающая, то есть для любого $n > 1$ выполняется $a_n > a_{n-1}$;
3. Для любого $n > 1$, a_n — наименьшее положительное число такое, что $a_{a_n} = n$.

Спешите! У вас есть всего несколько часов для того, чтобы помочь доктору Монту!

Формат входного файла

В единственной строке входного файла задано натуральное число n ($1 \le n \le 10^9$) — порядковый номер элемента в последовательности, который требуется Михаилу Афанасьевичу Монту.

Формат выходного файла

В единственной строке выходного файла выведите a_n.

Примеры

sequence.in	sequence.out
1	1

Задача F. Носки

Имя входного файла:	`socks.in`
Имя выходного файла:	`socks.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пришла весна и дед Олег решил пойти на свои грядки и заняться уборкой территории. Как и у многих представителей мужского пола, у него возникла трудность с поиском носок. Более того, почти все его носки оказались дырявыми. Олег перебрал всю свою кипу носок, и выписал размер носка и количество дырок на нём. Теперь он хочет собрать по паре носок каждого размера, который у него есть. При этом конечно же он хочет, чтобы в собранных парах было как можно меньше дырок.

Формат входного файла

В первой строке входного файла содержится положительное целое число N ($N \leq 299999$) — число обнаруженных носок. Дед Олег накопил большое количество носок, за время своего существования. В следующих N строках содержится описание каждого носка: его размер s_i и число дырок h_i ($1 \leq s_i \leq 10^9, 0 \leq h_i \leq 10^9$) — Дед Олег сам недоумевает, откуда у него могли оказаться носки размера 10^9, и откуда он мог получить такое большое количество дырок.

Формат выходного файла

В выходной файл выведите -1, если для какого-нибудь из имеющихся размеров нельзя найти пару носок (это бывает, когда остался только один носок данного размера), иначе выведите K — количество размеров носок, далее выведите K пар чисел, где каждая пара — номера носок одного размера с наименьшим количеством дырок. Носки нумеруются в том порядке, в котором они располагаются во входном файле $(1, 2, \ldots, N)$.

Примеры

socks.in	socks.out
10 29 2 29 3 27 1 27 2 29 4 29 3 31 2 31 0 31 0 27 3	3 3 4 1 2 8 9

Примечание

В примере Олег получает пару размера 27 с тремя дырками (носки 3 и 4), пару размера 29 с пятью дырками (носки 1 и 2) и пару размера 31 без дырок (носки 8 и 9).

Задача G. Старый стек

Имя входного файла:	`stack.in`
Имя выходного файла:	`stack.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Валера нашел в своем компьютере старую программу, реализующую структуру данных под названием стек. Напомним, что эта структура может быть представлена в виде высокого стакана, в который сверху можно добавлять элементы и сверху же из стека их доставать. Таким образом, стек поддерживает две операции: добавление элемента и удаление элемента. При этом элемент, добавленный в стек последним, будет извлечен из стека первым.

Однако стек в программе Валеры оказался настолько старым, что в нем образовались дыры, в которые элементы могут самопроизвольно вываливаться. Более формально, в стеке есть N дыр, каждая из которых находится на высоте h_i, а элемент в неё вываливается, если над ним находится минимум a_i элементов. Все элементы вываливаются одновременно (то есть если после добавления очередного элемента несколько элементов должны вывалиться в дыру, то они вывалятся все).

Помогите Валере по описанию стека и списку операций над ним понять, сколько элементов будет безвозвратно утеряно через дыры в стеке.

Формат входного файла

Первая строка входного файла содержит два целых числа N и $M (1 \le N \le 10^5, 1 \le M \le 3 \times 10^5)$ — количество дыр в стеке и количество операций, соответственно. В последующих N строках находится по два числа — $1 \le a_i \le 3 \times 10^5$ и $1 \le h_i \le 3 \times 10^5$ — описание i-той дыры в стеке. В следующей строке содержится M символов `+` и `-`. Символ `+` обозначает операцию добавления элемента в стек, символ `-` — операцию удаления.

Формат выходного файла

В выходной файл выведите единственное число — количество предметов, вывалившихся через дыры.

Примеры

stack.in	stack.out
2 4 1 1 2 1 +++-	2
2 5 1 2 2 1 +++++	4

Примечание

Рассмотрим первый пример:

1. Добавление. В стеке один элемент.
2. Добавление. В стеке два элемента. Один элемент вываливается в дыру на высоте 1 (поскольку над ним оказывается один элемент). Остается один элемент.
3. Добавление. Так же, как и на прошлом шаге, один элемент вываливается. Остается один элемент.
4. Удаление.

Таким образом, потеряется два элемента.

Рассмотрим второй пример:

1. Добавление. В стеке один элемент.
2. Добавление. В стеке два элемента. Элемент в дыру на высоте 1 не вываливается, поскольку над ним только один элемент, а в описании дыры указано, что требуется не менее двух.
3. Добавление. В стеке три элемента. Вываливается элемент в дыру на высоте 1 (потому что над ним оказывается два элемента). И вываливается элемент в дыру на высоте 2 (потому что над ним один элемент, а вываливаются элементы одновременно). В стеке остаётся один элемент.
4. Так же, как на шаге 2.
5. Так же, как на шаге 3.

Таким образом, потеряется четыре элемента.

Задача H. Фермерское чтиво

Имя входного файла:	`tree.in`
Имя выходного файла:	`tree.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Егор — очень почетный фермер. Довольные покупатели прозвали его «Пышный». Секрет его успеха, конечно же лежит в его манере управлять фермой. Например, у него организована строгая иерархия среди всего скота.

Более конкретно, всего в подчинении Егора находится $N - 1$ скотина с номерами от 2 до N, Егор не любит отбиваться от коллектива, поэтому тоже имеет номер – 1. У каждой скотины есть непосредственный начальник, причем ровно один. Начальником может быть либо другая скотина, либо сам Егор.

Недавно Егор решил постричь тех скотин, у которых нет подчиненных. Для начала ему нужно составить порядок в котором он будет их стричь.

Однако, появилось некоторое затруднение. Чтобы никого не обидеть, Егор должен учесть M требований вида: скотина, которую Егор будет стричь a_i-м по счету, должна находиться в подчинении (не обязательно прямом) скотины с номером b_i (при этом в требовании b_i может оказаться номер скотины без подчиненных, это значит, что именно эту скотину нужно стричь a_i-й по счету).

Теперь Егору интересно, сколько существует различных способов постричь скот, удовлетворив всем требованиям. Два способа считаются различными, если порядок стрижки отличается хотя бы в одном номере. Поскольку ответ может быть очень большим, выведите результат по модулю $10^9 + 7$.

Формат входного файла

Первая строка входного файла содержит два целых числа N и M ($2 \le N \le 3 \times 10^5, 0 \le M \le 3 \times 10^5$) — количество скотин и количество требований, соответственно. Скотина с номером 1 — Егор. В каждой i-й строке из последующих $N - 1$ строк находится единственное число $1 \le p_i < i + 1$ — начальник скотины с номером $i + 1$. В каждой i-й строке из последующих M строк находится по два числа $1 \le a_i, b_i \le 3 \times 10^5$ — скотина, которую будут стричь a_i-й, должна находится в подчинении (не обязательно прямом) скотины с номером b_i. Гарантируется, что a_i не превосходит количества скотин без подчиненных. Также номер b_i может принадлежать скотине без подчиненных, это означает, что эту скотину нужно стричь a_i-й по счету.

Формат выходного файла

В выходной файл выведите единственное число — количество различных способов стрижки по модулю $10^9 + 7$.

Примеры

tree.in	tree.out
5 3 1 1 2 2 1 2 2 1 3 2	2
6 1 1 1 2 2 2 4 3	6

Примечание

В первом примере требуется постричь скотин с номерами 3, 4 и 5. При этом скотины 4 и 5 находятся в подчинении у скотины номер 2 и при этом все скотины прямо или косвенно подчиняются Егору. Требования говорят, что в первую и третью очередь нужно стричь подчиненных номера 2. Таким образом мы можем либо в первую очередь стричь четвёртого, в третью пятого, либо наоборот. Третьего же остаётся стричь только во вторую очередь. Таким образом получаем два возможных порядка стрижки: `4 3 5` и `5 3 4`.