

Задача А. Наибольший общий делитель

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

Сегодня на уроке математики шестиклассник Петя изучил понятие наибольшего общего делителя. Петя тут же решил применить полученные знания на практике.

Петя выписал на листке бумаги n чисел a_1, \dots, a_n — номера домов, в которых живут его друзья. Теперь он хочет выбрать такое подмножество этих чисел, чтобы их наибольший общий делитель был равен его любимому числу d .

Помогите Пете выбрать из выписанных чисел искомое подмножество.

Формат входного файла

Первая строка входного файла содержит два целых числа n и d ($1 \leq n \leq 1000$, $1 \leq d \leq 10^9$). Вторая строка содержит n целых чисел: a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Формат выходного файла

Если существует искомое подмножество, выведите на первой строке выходного файла число k — количество чисел в нем. На второй строке выведите числа, входящие в это подмножество.

Если решения не существует, выведите на первой строке выходного файла число -1 .

Если возможных ответов несколько, выведите любой из них.

Примеры

gcd.in	gcd.out
4 3 6 8 12 9	2 6 9
3 3 2 4 8	-1

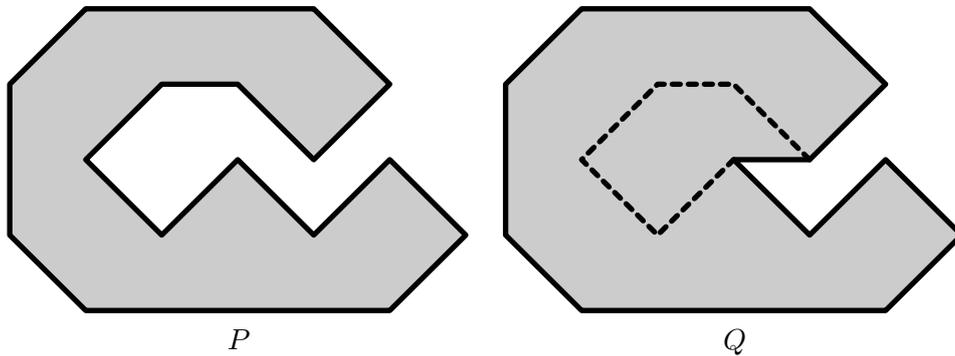
Задача В. Защита беженцев

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

Фридрих III — мэр обыкновенного средневекового города. Недавно на этот город стали совершать набеги кочевники. Город окружен надежной крепостной стеной, поэтому коренные жители города надежно защищены. К сожалению, бесчинство кочевников привело к наплыву беженцев, которые стали селиться у крепостной стены за пределами города. Именно они больше всего страдают от набегов. Фридрих считает, что защита беженцев входит в его обязанности, но в данный момент у города нет ресурсов на строительство дополнительной стены.

Оборона города на высоте, поэтому кочевники не рискуют приближаться вплотную к городу. Мэр заметил, что за пределами города существуют точки, которые не видны кочевникам до тех пор, пока те остаются достаточно далеко.

Более формально, представим крепостную стену как многоугольник P . Назовем точку X защищенной многоугольником P , если любой луч, проведенный из X , пересекается с P . Например, любая точка внутри P является защищенной. Множество всех защищенных точек также образует многоугольник. Назовем его Q .



Мэр решил распространить среди беженцев карту защищенных точек, как мест, рекомендуемых для поселения. Требуется помочь ему составить эту карту.

Задан многоугольник P , требуется найти многоугольник Q , состоящий из точек, защищенных многоугольником P .

Формат входного файла

Первая строка входного файла содержит целое число n — число вершин многоугольника P , соответствующего крепостной стене ($3 \leq n \leq 100$). Следующие n строк содержат по два целых числа x_i и y_i , разделенных пробелом — координаты i -й вершины многоугольника в порядке обхода по часовой стрелке ($|x_i|, |y_i| \leq 10\,000$).

Формат выходного файла

Выведите множество защищенных точек — многоугольник Q .

Сначала выведите число m — число вершин многоугольника Q . Следующие m строк должны содержать по два вещественных числа x_i и y_i , разделенных пробелом — координаты i -й вершины многоугольника в порядке обхода по часовой стрелке. Никакие три последовательные вершины не должны лежать на одной прямой.

Вещественные числа выводите с точностью не менее шести знаков после десятичной точки.

Примеры

guard.in	guard.out
3	3
0 0	0 0
0 1	0 1
1 0	1 0
16	12
0 1	0 1
0 3	0 3
1 4	1 4
4 4	4 4
5 3	5 3
4 2	4 2
3 3	3 2
2 3	4 1
1 2	5 2
2 1	6 1
3 2	5 0
4 1	1 0
5 2	
6 1	
5 0	
1 0	

Примечание

Второй пример соответствует рисунку, приведенному в условии задачи.

Задача С. Телефонный номер

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

Недавно Васе продиктовали номер телефона. Он записал его, разделив числа дефисами, но потом начал сомневаться, не ошибся ли он в записи номера. Ведь бывают различные телефонные номера, которые произносятся одинаково. Например, все четыре номера «3000-100-1», «3000-101», «3100-1», «3101» читаются как «три тысячи сто один». Теперь он хочет выяснить, какие номера произносятся также, как и записанный им номер.

Вася считает, что при произношении телефонного номера не употребляются числа, состоящие более чем из девяти цифр. Напомним, как произносятся числа от 1 до 999 999 999.

Если число больше или равно 1 000 000, то сначала произносится число миллионов в мужском роде, затем слово «миллион», «миллиона» или «миллионов», по следующим правилам. Если число миллионов, взятое по модулю 100, не равно 11 и дает остаток 1 по модулю 10, то произносится слово «миллион». Если число миллионов, взятое по модулю 100, не равно 12, 13 или 14 и дает остаток 2, 3 или 4 по модулю 10, то произносится слово «миллиона». Иначе произносится слово «миллионов».

Если число по модулю 1 000 000 больше или равно 1 000, то затем произносится число тысяч в женском роде, затем слово «тысяча», «тысячи» или «тысяч», по следующим правилам. Если число тысяч, взятое по модулю 100, не равно 11 и дает остаток 1 по модулю 10, то произносится слово «тысяча». Если число тысяч, взятое по модулю 100, не равно 12, 13 или 14 и дает остаток 2, 3 или 4 по модулю 10, то произносится слово «тысячи». Иначе произносится слово «тысяч».

Затем, если число по модулю 1 000 отлично от нуля, в мужском роде произносится число единиц.

Например, число 978 121 014 произносится как «девятьсот семьдесят восемь миллионов сто двадцать одна тысяча четырнадцать», а число 1 000 142 как «один миллион сто сорок два». Заметим, что перед словом «миллион» нельзя опускать слово «один», перед словом «тысяча» слово «одна», а числа от 11 до 19 произносятся одним словом.

Помогите Васе найти все телефонные номера, которые произносятся так же, как тот, который он записал.

Формат входного файла

На входе единственная строка, содержащая номер записанного Васей телефона. Номер телефона — это целые числа, разделенные символом «-» (ASCII код 39). Каждое число в записи положительно, содержит не более девяти цифр и не содержит ведущих нулей. Гарантируется, что в записанном Васей номере встречается не больше 50 цифр.

Формат выходного файла

В первой строке выведите количество телефонных номеров, которые читаются так же, как и записанный Васей номер. В следующих строках выведите эти номера телефонов. Каждый номер должен находиться в отдельной строке. Каждый номер должен состоять из чисел от 1 до 999 999 999, разделенных дефисами.

Порядок телефонов в ответе не важен. Гарантируется, что количество номеров в ответе не превышает 100 000. Учтите, что в ответе могут встречаться и номера, составленные более чем из 50 цифр.

Пример

<code>hear.in</code>	<code>hear.out</code>
3000-101	4 3000-100-1 3000-101 3100-1 3101

Задача D. Гостиница

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

На Всероссийскую командную олимпиаду школьников по программированию приезжает множество делегаций из различных городов нашей страны. Расселить делегации по номерам в гостинице является непростой задачей.

Из одного крупного города приехала делегация, состоящая из n человек. В гостинице, куда решено было заселить делегацию, имеются лишь двухместные и трехместные номера. Для экономии средств делегация хочет занять как можно меньше номеров, при этом в занимаемых номерах не должно оставаться свободных мест.

Помогите определить, каким образом можно разместить делегацию из n в двухместных и трехместных номерах, чтобы использовать суммарно минимальное число номеров.

Формат входного файла

В входном файле содержится единственное целое число n ($2 \leq n \leq 100$) — размер делегации.

Формат выходного файла

В выходной файл выведите два целых числа a_2 и a_3 , разделенных пробелом — число двухместных и трехместных номеров, которые необходимо выделить делегации, соответственно.

Примеры

<code>hotel.in</code>	<code>hotel.out</code>
7	2 1

Задача Е. Парад

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

Генералу Тупикову пришла в голову гениальная идея эффектного шоу во время проведения парада в честь пятидесятилетия победы в маленькой победоносной войне Флатландии над Берляндией.

В параде примет участие n солдат. Солдаты, выстроенные в колонну по одному, выйдут на главную площадь. Затем, по команде генерала, некоторые солдаты сделают шаг влево, а остальные солдаты — шаг вправо. В результате солдаты, сделавшие шаг влево, должны оказаться упорядочены по возрастанию роста, а сделавшие шаг вправо — по убыванию.

Генерал приказал немедленно принести ему список солдат, которые примут участие в параде, чтобы определить, в каком порядке им следует построиться, кому сделать шаг влево, а кому — вправо. Однако выяснилось, что репетиции парада продолжаются уже несколько месяцев и солдаты знают, в каком порядке они должны выйти на площадь.

Эта новость слегка огорчила генерала, но он не упал духом. Он выяснил рост всех солдат, которые примут участие в параде, и теперь хочет разбить всех солдат на два непустых множества: тех кто сделает шаг влево и тех, кто сделает шаг вправо. Сделавшие шаг влево должны оказаться упорядочены по возрастанию роста, а сделавшие шаг вправо — по убыванию.

Помогите генералу составить сценарий парада, выясните, кому из солдат надо отдать команду сделать шаг влево. Остальные солдаты сделают шаг вправо. Хотя бы один солдат должен сделать шаг влево, и хотя бы один солдат должен сделать шаг вправо.

Формат входного файла

В первой строке входного файла задано целое число n ($2 \leq n \leq 100\,000$) — количество солдат. Во второй строке задано n целых положительных чисел a_1, a_2, \dots, a_n — рост солдат ($1 \leq a_i \leq 10^9$). Рост солдат приведен в порядке, в котором они выйдут на площадь.

Формат выходного файла

В первой строке выходного файла выведите k — количество солдат, которым надо сделать шаг влево ($1 \leq k \leq n - 1$). В второй строке выходного файла выведите k целых чисел b_1, b_2, \dots, b_k — номера солдат, которым надо сделать шаг влево ($1 \leq b_1 < b_2 < \dots < b_k \leq n$).

Рост этих солдат должен строго возрастать, а рост оставшихся солдат должен строго убывать. Если решений несколько, выведите любое.

В случае, если решения нет, в единственной строке выходного файла выведите «Impossible».

Примеры

parade.in	parade.out
6 6 1 4 3 2 5	3 2 4 6
6 1 3 2 4 6 5	Impossible
3 1 1 1	Impossible

Задача F. Магазин

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

У Билла большая семья: трое сыновей, девять внуков. И всех надо кормить. Поэтому Билл раз в неделю ходит в магазин.

Однажды Билл пришел в магазин и увидел, что в магазине проводится акция под названием «каждый k -й товар бесплатно». Изучив правила акции, Билл выяснил следующее. Пробив на кассе товары, покупатель получает чек. Пусть в чеке n товаров, тогда n/k округленное вниз самых дешевых из них достаются бесплатно.

Например, если в чеке пять товаров за 200, 100, 1000, 400 и 100 рублей, соответственно, и $k = 2$, то бесплатно достаются оба товара по 100 рублей, всего покупатель должен заплатить 1600 рублей.

Билл уже выбрал товары, и направился к кассе, когда сообразил, что товары, которые он хочет купить, можно разбить на несколько чеков, и благодаря этому потратить меньше денег.

Помогите Биллу выяснить, какую минимальную сумму он сможет заплатить за выбранные товары, возможно разбив их на несколько чеков.

Формат входного файла

Первая строка входного файла содержит два целых числа n, k ($1 \leq n \leq 100\,000, 2 \leq k \leq 100$) — количество товаров, которые хочет купить Билл и параметр акции «каждый k -й товар бесплатно».

Следующая строка содержит n целых чисел a_i ($1 \leq a_i \leq 10\,000$) — цены товаров, которые покупает Билл.

Формат выходного файла

Выведите в выходной файл одно число — минимальную сумму, которую должен заплатить Билл за товары.

Примеры

<code>shop.in</code>	<code>shop.out</code>
5 2 200 100 1000 400 100	1300

В приведенном примере Билл может разбить товары на два чека: в один чек пойдут товары за 1000 и за 400 рублей, товар за 400 рублей в этом чеке достанется бесплатно, а в другой — остальные товары, в нем бесплатно достанется один товар за 100 рублей. Итого Биллу придется заплатить 1300 рублей.

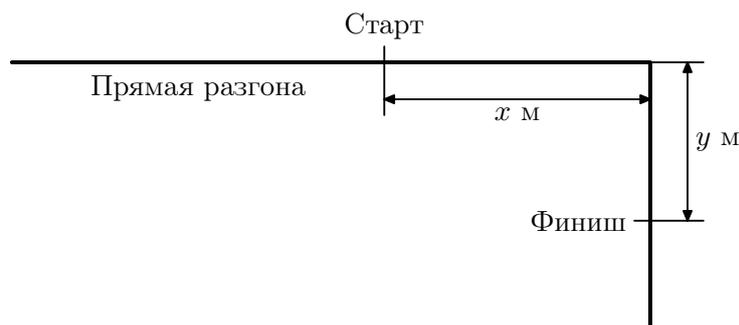
Задача G. Занос

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

В Берляндии готовятся к проведению очередного этапа по гонкам машин класса I2011.

Известно, что машины этого класса не могут двигаться со скоростью, превышающей v м/с, при разгоне мощности двигателя хватает, чтобы развить любое ускорение, не превышающее a м/с², а при торможении абсолютная величина ускорения не превышает b м/с².

Трасса в Берляндии устроена следующим образом. Сначала идет длинная прямая, на которой машины могут разогнаться до любой скорости от 0 до v . Затем следует линия старта, после которой следует отрезок длиной x м, поворот на 90° и отрезок длины y м, после которого находится линия финиша. Линию финиша машина может пересечь на любой скорости.



Для прохождения поворота рекомендуется использовать управляемый занос. Для этого необходимо затормозить таким образом, чтобы скорость оказалась равна 0 в точности в повороте. После этого, сделав резкое движение рулем по направлению поворота и резко нажав газ, можно, не теряя времени, уйти на второй отрезок трассы.

Виталий планирует участвовать в соревнованиях. Он решил рассчитать, за какое минимальное время он сможет пройти трассу. Помогите ему это сделать.

Формат входного файла

Входной файл содержит пять целых чисел: v, x, y, a, b ($1 \leq v, x, y, a, b \leq 10^6$).

Формат выходного файла

Выведите в выходной файл минимальное время, за которое возможно пройти всю трассу. Ваш ответ должен иметь абсолютную или относительную погрешность не больше 10^{-8} .

Примеры

<code>sideslip.in</code>	<code>sideslip.out</code>
4 10 12 1 2	8.5

В приведенном примере следует действовать так. На прямой разгона следует разогнаться до максимальной возможной скорости в 4 м/с. После старта следует проехать 6 м на максимальной возможной скорости за 1.5 с, затем за 2 с затормозить до 0, используя максимальное возможное торможение, заодно проехав оставшиеся 4 м до поворота. Войдя с заносом в поворот, следует нажать на газ и с максимальным ускорением разогнаться за 4 с до максимальной скорости в 4 м/с, пройдя за это время 8 м. После этого оставшиеся 4 м до финиша следует пройти за 1 с на максимальной скорости. Общее время на прохождение трассы равно $1.5 + 2 + 4 + 1 = 8.5$ с.

Задача Н. Чай

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

Члены жюри Всероссийской командной олимпиады школьников по программированию очень любят пить чай. Но, к сожалению, в их распоряжении всего один чайник, имеющий объем v мл. Работа жюри хорошо отлажена, поэтому точно известно, кто и когда отправиться пить чай, и сколько воды ему для этого нужно.

Рассмотрим действия, предпринимаемые членами жюри для получения желанного напитка.

Когда человек из жюри подходит к чайнику, то если никого больше у чайника нет, он смотрит, достаточно ли в чайнике воды. Если в чайнике не хватает воды для этого человека, то он доликает в чайник воды до полного объема в v мл. Если же воды достаточно, то вода в чайник не доливается. После этого чайник включается и доводит воду до температуры 100 градусов. После этого член жюри наливает себе необходимый объем кипятка для заварки чая и отходит от чайника.

Если в момент прихода к чайнику член жюри обнаруживает других людей, ожидающих кипятка, то он становится в очередь и дожидается, пока все предыдущие люди выполнят свои действия с чайником.

Температура кипения воды составляет 100 градусов Цельсия. Мощность чайника составляет N , что означает, что если в чайнике x мл воды, то температура воды в чайнике изменяется по формуле $T(t) = T_0 + \frac{Nt}{x}$, где T_0 — температура воды в момент включения чайника, а t — время в секундах, прошедшее с момента включения чайника. Когда температура воды достигает 100 градусов, чайник автоматически выключается.

Когда чайник выключен, вода в нем начинает остывать. Вне зависимости от количества жидкости в чайнике, температура воды падает со скоростью k градусов в секунду. Когда вода достигает комнатной температуры в 20 градусов, остывание прекращается.

Заливаемая в чайник вода имеет комнатную температуру 20 градусов. При доливании воды в чайник температура воды усредняется, а именно, если в чайнике было w мл воды температуры T , то температура воды в чайнике после его заполнения станет $\frac{wT+20(v-w)}{v}$.

Известны времена, когда члены жюри идут пить чай, а также количество кипятка им необходимое. Требуется для каждого члена жюри вывести в какой момент он начнет пить чай. Считается, что разливают воду по чашкам и заваривают чай члены жюри моментально, при этом всегда наливая себе ровно необходимый им объем. Изначально чайник пуст.

Формат входного файла

Первая строка входного файла содержит четыре целых числа m , v , N и k ($1 \leq m \leq 100\,000$, $1 \leq v, N, k \leq 1000$) — количество членов жюри, пьющих чай, объем чайника в мл, мощность чайника и скорость остывания воды в чайнике. Далее следуют m строк — описания людей, пьющих чай. В каждой строке два целых числа t_i и a_i ($1 \leq t_i \leq 10^6$; $1 \leq a_i \leq v$) — время в секундах от начала олимпиады, когда член жюри отправиться пить чай, и необходимый ему объем воды в мл. Никакие два члена жюри не отправляются пить чай одновременно.

Формат выходного файла

В выходной файл выведите m строк по одному числу в каждой — время в секундах от начала олимпиады, когда соответствующий член жюри начнет пить чай. Ответ должен иметь абсолютную или относительную погрешность не более 10^{-7} .

Примеры

tea.in	tea.out
3 1000 200 1	401.0
1 500	701.0
501 300	1021.0
551 300	

Задача I. Командная олимпиада

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

Совсем скоро начнется первый тур очередной всероссийской командной олимпиады школьников по палеонтологии (ВКОШП). На олимпиаду приехали команды из n школ, от каждой школы приехало ровно по две команды. Команды уже заняли свои места, когда обнаружилось, что некоторые команды из одной школы сидят слишком близко. Перед организаторами олимпиады встала серьезная задача — пересадить участников олимпиады.

Столы, за которыми сидят команды, расставлены в один ряд. Расстояние между рабочими местами соседних команд оказалось равно 10 метрам. Организаторы хотят, чтобы минимальное расстояние между двумя командами из одной школы было как можно больше.

Пересаживая команду, организатором необходимо перенести на новое место все оборудование, приготовленное командой для исследований. Поэтому организаторы хотят пересадить команды так, чтобы сумма расстояний между старыми и новыми рабочими местами команд была как можно меньше.

Например, пусть в соревновании принимают участие по две команды школ 1, 2, 3 и 4. Пусть исходно команды распределены по рабочим местам следующим образом: 1, 3, 2, 2, 1, 4, 4, 3 (для каждой команды указан номер школы, которую она представляет). При таком распределении по рабочим местам команды из школы 2 сидят слишком близко, как и команды из школы 4. Пересадив команды в следующем порядке: 1, 3, 2, 4, 1, 3, 2, 4, жюри может добиться своего: команды из одной школы сидят на местах, расстояние между которыми не меньше 40 м, большего расстояния добиться нельзя. Сумма расстояний между старыми и новыми позициями для данного примера равна $0 + 0 + 0 + 20 + 0 + 20 + 30 + 10 = 80$ м, для исходного распределения команд она минимальна.

Вам задано исходное распределение команд по рабочим местам. Требуется пересадить их так, чтобы минимальное расстояние между командами из одной школы было как можно больше. При этом среди возможных новых размещений команд следует выбрать такое, чтобы сумма расстояний между старыми и новыми местами рабочими команд была минимально возможной.

Формат входного файла

В первой строке входного файла задано число n — количество команд ($1 \leq n \leq 100$). Во второй строке задано исходное распределение команд по рабочим местам — последовательность a_1, a_2, \dots, a_{2n} . Для каждой команды указан номер школы, которую она представляет. Гарантируется, что последовательность состоит из чисел от 1 до n и каждое число встречается ровно два раза

Формат выходного файла

В единственную строку выходного файла выведите, каким образом следует пересадить команды, чтобы минимальное расстояние между командами из одной школы было как можно больше. При этом среди возможных новых размещений команд следует выбрать такое, чтобы сумма расстояний между старыми и новыми рабочими местами команд была минимальной возможной. Если оптимальных ответов несколько, можно вывести любой из них.

Примеры

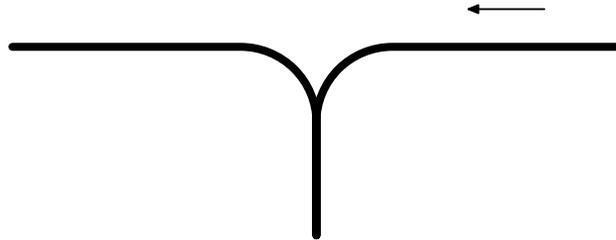
<code>teams.in</code>	<code>teams.out</code>
4	1 3 2 4 1 3 2 4
1 3 2 2 1 4 4 3	

Задача J. Поезда

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

Маленький Вася очень любит поезда. На день рождения ему подарили игрушечную железную дорогу. В комплект к железной дороге входит поезд, состоящий из n вагонов, пронумерованных числами от 1 до n .

Любимым занятием Васи стала сортировка поездов с использованием специального *сортировочного тупика*.



Справа к тупику подъезжает поезд, составленный из всех n вагонов. Затем вагоны по одному загоняются в сортировочный тупик и выгоняются из него налево. Тупик достаточно длинный и в нем может одновременно поместиться любое количество вагонов.

Васе нравится, если ему удастся отсортировать поезд с помощью сортировочного тупика — добиться того, чтобы слева от тупика вагоны были расположены по порядку от 1 до n .

Например, пусть в исходном поезде 4 вагона, которые следуют в порядке 3, 1, 2, 4. Его можно отсортировать следующим образом. Загоняем вагон 3 в тупик. Загоняем вагон 1 в тупик. Выгоняем вагон 1 из тупика. Загоняем вагон 2 в тупик. Выгоняем вагон 2 из тупика. Выгоняем вагон 3 из тупика. Загоняем вагон 4 в тупик. Выгоняем вагон 4 из тупика.

Не все поезда можно отсортировать таким образом. Например, поезд из 3 вагонов, следующих в порядке 2, 3, 1, отсортировать нельзя.

Вася выписал на листке в лексикографическом порядке все поезда длины n , которые можно отсортировать с помощью тупика. Поезд (a_1, a_2, \dots, a_n) идет раньше поезда (b_1, b_2, \dots, b_n) в лексикографическом порядке, если существует такое i ($1 \leq i < n$), что для всех $j < i$ выполняется $a_j = b_j$, а $a_i < b_i$. Например, все поезда из трех вагонов, которые можно отсортировать с помощью тупика, в лексикографическом порядке выписываются следующим образом: (1, 2, 3), (1, 3, 2), (2, 1, 3), (3, 1, 2), (3, 2, 1).

Вася потерял свой листок, и его интересует вопрос: какой поезд был выписан на его листке под номером k . Помогите ему выяснить это.

Формат входного файла

Входной файл содержит два целых числа — n и k ($1 \leq n \leq 30$, $1 \leq k \leq 10^{18}$).

Формат выходного файла

Выведите в выходной файл n целых чисел: k -й в лексикографическом порядке поезд из n вагонов, который можно отсортировать с помощью тупика.

Если с помощью тупика можно отсортировать менее k поездов из n вагонов, выведите -1 .

Примеры

<code>trains.in</code>	<code>trains.out</code>
4 1	1 2 3 4
4 3	1 3 2 4
4 15	-1

Задача К. Королевская династия

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

Глеб очень любит историю. Он приходит в полный восторг, когда ему предлагают изучить историю какой-либо конкретной династии. Глеб очень строго относится к чистоте крови, поэтому при анализе династии он включает в рассмотрение только мужчин, которые являются кровными потомками основателя династии по отцовской линии.

Один из главных методов исторического анализа династии — изучение количества сыновей некоторого ее представителя. Глеб же намеревается произвести настоящую революцию в исследованиях — он намерен изучать не просто количество сыновей, а количество внуков, правнуков, праправнуков и так далее. Однако, династии могли тянуться несколько десятков поколений, а значит общее число кровных потомков очень велико, так что Глебу стало очень тяжело работать. Поскольку на дворе сейчас XXI век, Глеб решил обратиться за помощью к квалифицированным программистам.

Династия представляет собой множество людей, один из которых называется основателем династии, а любой другой представитель династии U имеет отца V , являющегося представителем династии. При этом U является сыном V , или *потомком U через 1 поколение*. Потомком U через $k + 1$ поколение называется сын V некоторого потомка U через k поколений.

Глеба интересует информация о том, сколько у некоторого представителя династии V существует потомков через k поколений. Разумеется, Глеба интересует далеко не один такой вопрос.

Формат входного файла

В первой строке входного файла содержится одно число n — количество человек в династии, которую исследует Глеб ($1 \leq n \leq 10^5$), представители династии пронумерованы различными натуральными числами от 1 до n . Далее следуют n чисел, i -е из которых задает номер отца i -го представителя династии, или равно -1 , если соответствующий представитель является основателем династии.

Гарантируется, что основатель династии ровно один, и что любой упомянутый представитель династии является потомком основателя династии.

В следующей строке содержится число m — количество вопросов, которое интересует Глеба ($1 \leq m \leq 10^5$). Далее следует m строк, каждая из которых содержит два целых числа v и k ($1 \leq v \leq n$, $1 \leq k \leq 10^9$) — номер исследуемого представителя династии и интересующее Глеба число поколений.

Формат выходного файла

Для каждого вопроса выведите одно число — количество потомков через соответствующее число поколений у заданного представителя династии.

Примеры

<code>tree.in</code>	<code>tree.out</code>
5	2
-1 1 2 1 4	0
2	
1 2	
4 7	

В первом запросе у представителя династии номер 1 есть два потомка во 2 поколении: 3 и 5.