

Задача А. Аааааааа

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

Задана строка, состоящая из строчных латинских букв. Рассмотрим следующую операцию: выбрать два соседних индекса i и $i + 1$ и одновременно сдвинуть два соседних символа по циклу на один. Например, символ «а» перейдёт в «b», «b» — в «c», ..., «z» — в «a».

Назовём строку хорошей, если такими преобразованиями из неё можно получить строку такой же длины, состоящую только из букв «а».

Вам дана строка s . Требуется ответить на q запросов, в которых спрашивается является ли некоторая подстрока строки s хорошей.

Формат входных данных

В первой строке задана строка s ($1 \leq |s| \leq 1\,000\,000$). Строчка s содержит только строчные латинские буквы от «а» до «z».

Вторая строка содержит число запросов q ($1 \leq q \leq 1\,000\,000$).

Каждая из следующих q строк содержит числа l и r ($1 \leq l \leq r \leq |s|$) — левая и правая границы отрезка очередного запроса.

Формат выходных данных

Для каждого запроса в отдельной строке выведите «Yes», если подстрока является хорошей, и «No», иначе.

Пример

стандартный ввод	стандартный вывод
abaabcb	No
5	Yes
1 3	No
3 4	No
4 5	Yes
4 6	
5 7	

Замечание

В примере из условия подстрока «aa» из второго запроса уже является хорошей.

Подстрока из последнего запроса преобразуется следующим образом: «bcb» → «abb» → «aaa».

Для всех остальных запросов получить слово, состоящее только из букв «а», невозможно.

Задача В. Бейджик

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

Как вы, возможно, знаете в ЛКШ за плохое поведение ставят дырки в бейджик. Вот и сегодня один из суровых преподавателей поймал группу из n школьников за очередной проделкой. Пронумеруем для удобства школьников в этой группе от 1 до n .

Преподаватель подошёл к школьнику a и поставил ему дырку в бейджик. Тот, правда, попытался оправдаться, что зачинщиком бы некоторый другой школьник p_a .

После этого преподаватель подошёл к школьнику p_a и поставил дырку в бейджик и ему. Тот в ответ послал его к школьнику p_{p_a} .

Так продолжалось некоторое время, но, так как количество школьников было конечно, то через некоторое время преподаватель пришёл к школьнику, которому он уже поставил дырку.

После этого он поставил в бейдж этого школьника вторую дырку и счёл на этом свой долг выполненным.

Вы не знаете с какого школьника a начал преподаватель. Но вы знаете все числа p_i . Выясните для каждого возможного школьника a того школьника, которому была поставлена вторая дырка.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \leq n \leq 100\,000$) — количество нашкодивших школьников.

Вторая строка содержит n целых чисел a_1, \dots, a_n ($1 \leq a_i \leq n$), где a_i это номер школьника, к которому посылает преподавателя школьник под номером i .

Формат выходных данных

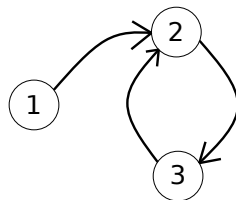
Выведите n чисел — номер школьника, которому поставят две дырки, для каждого школьника a от 1 до n .

Примеры

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

Замечание

Картинка иллюстрирует первый пример.



В первом примере, если $a = 1$, то преподаватель подойдёт к школьникам 1, 2, 3, 2, тем самым вторую дырку получит школьник 2.

В первом примере, если $a = 2$, то преподаватель подойдёт к школьникам 2, 3, 2 и школьник 2 получит вторую дырку в бейдж. Если $a = 3$, то учитель подойдёт к школьникам 3, 2, 3, тем самым школьник 3 получит вторую дырку в бейджик.

Во втором примере для любого a учитель подойдёт к школьникам a, a , тем самым вторую дырку получает школьник a .

Задача С. Сезон дождей

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

К 3018 году Летняя Компьютерная Школа довольно сильно увеличилась в размерах. Новым местом проведения был выбран отель «Берендеетроник». База состоит из n домиков, между которыми есть $n - 1$ тропинок, и между любыми домиками есть путь.

Все на базе было прекрасно, пока не начались дожди. Прогноз погоды обещает, что дожди будут идти m дней. Специальный отряд преподавателей смог вычислить, что по i -й тропинке, соединяющей некие домики u_i и v_i , до дождей можно пройти за b_i секунд. Дождь же сильно размывает тропинку, и с каждым днем путь будет занимать на a_i секунд больше, иными словами, в t -й (с нуля) день после начала дождя прохождение тропинки будет занимать $a_i \cdot t + b_i$ секунд.

К сожалению, несмотря на все прилагаемые усилия, даже в 3018 году не все школьники встречают вечерку и мягкий отбой в своих домиках. Поскольку к жесткому отбою все обязаны уснуть, необходимо вычислить максимальное время пути между всеми парами домиков для каждого дня, и вывесить его на доску объявлений, чтобы каждый школьник знал последний момент, когда он бежать к себе домой.

Посчитайте максимальные времена пути между парой домиков через $t = 0, t = 1, \dots, t = m - 1$ дней.

Формат входных данных

В первой строке даны два целых числа n и m — число домиков на базе и количество дождливых дней ($1 \leq n \leq 100\,000$; $1 \leq m \leq 1\,000\,000$).

В следующих $n - 1$ строках даны четверки чисел u_i, v_i, a_i, b_i — описания тропинок ($1 \leq u_i, v_i \leq n$; $0 \leq a_i \leq 10^5$; $0 \leq b_i \leq 10^9$). i -я тропинка соединяет домики u_i и v_i , и в день t требует $a_i \cdot t + b_i$ секунд на прохождение.

Гарантируется, что между любыми двумя домиками есть путь.

Формат выходных данных

Выведите m чисел — значения самого длинного пути по базе через $t = 0, t = 1, \dots, t = m - 1$ дней после начала дождя.

Пример

стандартный ввод	стандартный вывод
5 10	200 200 200 210 220 230 260 290 320
1 2 0 100	350
1 3 0 100	
1 4 10 80	
1 5 20 0	

Замечание

Рассмотрим подробнее первый пример.

В первые три дня ($0 \leq t \leq 2$) самый длинный путь проходит между вторым и третьим домиками, и его длина равна $100 + 100 = 200$.

В третий день ($t = 2$) дорожка между домиками 1 и 4 становится по длине равной 100, и продолжает увеличиваться. Поэтому в дни с номерами $t = 2, t = 3, t = 4, t = 5$ самый длинный путь проходит между вершинами 4 и 1 или 2, и имеет длину $180 + 10t$. Заметим, что в день $t = 2$ есть три тропинки длиной 100, и есть три максимальных пути одинаковой длины.

В шестой день ($t = 5$) тропинка между первым и пятым домиком по длине догоняет первые две, и становится равной 100. Таким образом, во все дни с $t = 5$ и далее самый длинный путь проходит между вершинами 4 и 5, и имеет длину $80 + 30t$.

Задача D. Выборы

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

Как известно, большинство учеников и преподавателей большую часть года живут в Берляндии, в которой достаточно широко распространена коррупция. Поэтому рассмотрим следующий жизненный сюжет.

Скоро должны пройти выборы. Вы знаете количество избирателей и количество партий — n и m соответственно. Также, для каждого избирателя вы знаете, за какую партию он собирается голосовать. Однако, при наличии определённой суммы денег, это всё поправимо. В частности, если заплатить i -му избирателю c_i байткоинов, можно сделать так, чтобы тот проголосовал за любую партию по вашему усмотрению.

Объединённая Партия Государства Берляндии заказала у вас статистическое исследование — вам нужно выяснить минимальное количество байткоинов, которое ей нужно потратить, чтобы гарантировать себе победу. Чтобы партия выиграла выборы, ей необходимо набрать строго больше голосов, чем набрала любая из остальных партий.

Формат входных данных

В первой строке входных данных записаны два целых числа n и m ($1 \leq n, m \leq 3000$) — количество избирателей и количество партий соответственно.

Каждая из следующих n строк содержит числа p_i и c_i ($1 \leq p_i \leq m, 1 \leq c_i \leq 10^9$) — номер партии, за которую собирается голосовать очередной избиратель, а второе — количество байткоинов, за которое он готов изменить своё мнение.

Объединённая Партия Государства Берляндии имеет номер 1.

Формат выходных данных

Выведите одно число — минимальное количество байткоинов, которое нужно заплатить избирателям, чтобы Объединённая Партия Государства Берляндии смогла победить.

Примеры

стандартный ввод	стандартный вывод
1 2 1 100	0
5 5 2 100 3 200 4 300 5 400 5 900	500
5 5 2 100 3 200 4 300 5 800 5 900	600

Замечание

В первом тесте из условия Объединённая Партия выигрывает выборы, не покупая голоса избирателей.

Во втором тесте из условия Объединённая Партия может перекупить голоса первого и четвёртого избирателя. Таким образом, она наберёт два голоса, партии номер 3, 4 и 5 по одному, партия номер 2 не получит ни одного голоса.

В третьем тесте из условия Объединённая Партия может купить голоса первых трёх избирателей и выиграть, набрав три голоса против двух голосов у пятой партии.

Задача Е. Новое здание для ЛКШ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Перед вами на столе план нового здания, где планируется проводить Летнюю Компьютерную Школу. Вы работаете над логистикой для ЛКШ, поэтому вам интересно, как быстро можно добраться между разными важными точками: например, сколько минут идти от аудитории до столовой или от спортзала до серверной.

В здании N башен по H этажей. Между соседними башнями (то есть башнями с номерами i и $i + 1$) есть переходы на всех этажах $x : A \leq x \leq B$. Подняться или спуститься на один этаж внутри башни можно ровно за одну минуту. Также за одну минуту можно перейти между соседними башнями, если на вашем этаже есть переход. Покидать здание не разрешается.

У вас есть список из K пар мест $(t_a, f_a), (t_b, f_b)$: этаж f_a башни t_a и этаж f_b башни t_b . Для каждой пары мест требуется вычислить минимальное время перемещения между ними.

Формат входных данных

В первой строке ввода записаны целые числа:

- N : количество башен в здании ($1 \leq N \leq 10^8$)
- H : количество этажей в башнях ($1 \leq H \leq 10^8$)
- $[A, B]$: этажи, где есть переходы между башнями ($1 \leq A \leq B \leq H$)
- K : количество запросов минимального времени перемещения ($1 \leq K \leq 10^5$)

В следующих K строках записано по четыре целых числа: t_a, f_a, t_b, f_b ($1 \leq t_a, t_b \leq N$, $1 \leq f_a, f_b \leq H$). Эти числа соответствуют запросу на поиск минимального времени перемещения между f_a -м этажом t_a -й башни и f_b -м этажом t_b -й башни.

Формат выходных данных

Для каждого запроса выведите одно целое число на отдельной строке: минимальное время перемещения.

Пример

стандартный ввод	стандартный вывод
3 6 2 3 3	1
1 2 1 3	4
1 4 3 4	2
1 2 2 3	

Задача F. Большой треугольник

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Есть странная особенность: если соединить между собой города Ростов, Таганрог, Шахты, то получится треугольник

«Невероятно, но факт»

В ЛКШ приезжают ученики из самых разных уголков России и зарубежья. Вы отметили на карте все города, откуда приехали участники ЛКШ.

Затем, вы решили подготовить интересную инфографику на основе этой карты. Первое что вы захотели сделать — это найти три города на этой карте, так чтобы получился треугольник площади ровно S .

Формат входных данных

В первой строке входных данных находится два целых числа n и S ($3 \leq n \leq 2000$, $1 \leq S \leq 2 \cdot 10^{18}$) — количество городов на карте и требуемая площадь треугольника.

В следующую n строках находится описание городов, по одной на строке. Каждый город описывается своими координатами x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$).

Гарантируется, что все города находятся в различных точках. Также гарантируется, что никакие три города не лежат на одной прямой.

Формат выходных данных

Если решения не существует — выведите «No» (без кавычек).

Иначе выведите «Yes» (без кавычек), после чего выведите три пары координат — координаты городов, образующих треугольник площади S .

Примеры

стандартный ввод	стандартный вывод
3 7 0 0 3 0 0 4	No
4 3 0 0 2 0 1 2 1 3	Yes 0 0 1 3 2 0

Задача G. Шляпа

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

Это интерактивная задача.

Имур Ихаков организует клуб по шляпе. В клубе решило принять участие чётное число человек. Имур усадил их всех в круг и провёл жеребьевку, чтобы разбить ребят на пары, но что-то пошло не так. Участники пронумерованы так, что i и $i + 1$ ($1 \leq i < n$) сидят рядом, а также участники n и 1 сидят рядом. Каждому был выдан листочек с числом так, что у ребят, сидящих рядом, эти числа отличаются ровно на единицу. Предполагалось, что игроки с одинаковыми числами образуют пару, но оказалось, что не все числа встречались ровно дважды.

Как известно, удобнее всего объяснять слова партнёру, когда он сидит напротив. Участники с номерами i и $i + \frac{n}{2}$ сидят напротив друг друга. Имуре интересно, есть ли люди, сидящие напротив с одинаковыми числами на своих листочках. Помогите ему найти такую пару людей, если она есть.

За один вопрос можно узнать число на листочке человека с номером i . Имур не любит отвечать на вопросы, поэтому их должно быть не больше 60.

Формат входных данных

На вход подается целое чётное число n ($2 \leq n \leq 100\,000$) — число участников на клубе Имюра. Вам разрешается задать не более 60 вопросов.

Формат выходных данных

Для того, чтобы узнать число i -го участника ($1 \leq i \leq n$), нужно вывести «? i ». После этого ваша программа на вход получит целое число a_i ($-10^9 \leq a_i \leq 10^9$) — число на листочке i -го человека.

Чтобы сообщить, что ответ найден, требуется вывести «! i », где i — номер любого участника из пары ($1 \leq i \leq n$). Если такой пары участников не существует, выведите «! -1». После этого программа должна завершиться.

Тестирующая система выдаст вашей программе прочитать ответ на запрос из входных данных только после того, как ваша программа вывела соответствующий запрос системе и выполнила операцию `flush`.

Примеры

стандартный ввод	стандартный вывод
8	? 4
2	? 8
2	! 4
6	? 1
1	? 2
2	? 3
3	? 4
2	? 5
1	? 6
0	! -1

Замечание

Ввод-вывод в примерах демонстрирует пример взаимодействия.

В первом примере были загаданы результаты жеребьёвки соответствующие последовательности 1, 2, 1, 2, 3, 4, 3, 2

Во втором примере была загадана последовательность 1, 2, 3, 2, 1, 0.

Задача Н. Ноутбуки

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

Как можно было догадаться, в Летней Компьютерной Школе нельзя обойтись без ноутбука. А когда есть много ноутбуков, то им нужно много зарядников. В частности, в этой смене ЛКШ есть n ноутбуков и m зарядников, причём любой зарядник может заряжать любой ноутбук. Один зарядник не может заряжать одновременно более одного ноутбука.

Ноутбук с номером i теряет a_i заряда за секунду. Если же он подключён зарядником к розетке, то он дополнительно заряжается на b_i заряда в секунду (изменение заряда за секунду составит $b_i - a_i$). Процесс зарядки и разрядки ноутбуков происходит не дискретно, а непрерывно. Например, если разряжать ноутбук половину секунды, то он потеряет $\frac{1}{2}a_i$ заряда.

Дополнительно, i -й ноутбук имеет ёмкость батареи c_i и изначально полностью заряжен.

Определите, можно ли поддерживать все ноутбуки включёнными (то есть с положительным уровнем заряда) бесконечно долго, если разрешается переключать зарядные устройства сколь угодно часто?

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 50\,000$) — количество ноутбуков и количество зарядников в ЛКШ.

Каждая из следующих n строк содержит три числа a_i, b_i, c_i ($1 \leq a_i \leq 1000, 1 \leq b_i \leq 32, 1 \leq c_i \leq 10^9$) — скорость разрядки, скорость зарядки и ёмкость батареи i -го ноутбука.

Формат выходных данных

Если можно поддерживать ноутбуки включёнными неограниченно долго, выведите «Yes», иначе выведите «No».

Примеры

стандартный ввод	стандартный вывод
2 1 1 2 30 1 2 30	Yes
2 1 3 3 10 2 2 10	No

Замечание

В первом тесте за одну секунду оба компьютера теряют 1 единицы заряда. Если мы будем заряжать каждый из них по полсекунды, то у обоих компьютеров уровень заряда не будет понижаться.

Во втором тесте никак не получится держать оба ноутбука включёнными сколь угодно долго, потому что одновременно можно поставить на зарядку не более одного ноутбука (причём заряжаемый ноутбук не увеличивает свой заряд, а просто сохраняет его на прежнем уровне), а другой из них теряет свой заряд, поэтому рано или поздно заряд одного из ноутбуков закончится.

Задача I. Граф, биты и AND

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф из n вершин и m ребер, а так же вершины a и b в нем.

Требуется найти максимальный путь из a в b , если вес пути определяется как побитовый AND весов, входящих в него ребер.

Формат входных данных

Первая строка входных данных содержит два натуральных числа n и m ($2 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$) — количество вершин и ребер графа соответственно.

Каждая из следующих m строк содержит описание очередного ребра. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 10^9$, $b_i \neq e_i$) — номера концов ребра и его вес, соответственно.

В последней строке входных данных даны два числа a и b ($1 \leq a, b \leq n$, $a \neq b$) — концы пути.

Формат выходных данных

Выведите одно целое число — длину максимального пути из a в b и «-1», если пути из a в b не существует.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 16 1 3 7 2 3 7 3 4 3 1 4	3
6 6 1 2 7 2 3 8 3 6 7 1 4 5 4 5 9 5 6 9 1 6	1
2 0 2 1	-1

Замечание

В первом тесте в любом случае придется пройти по ребру веса 3. Поэтому, если мы пройдем по ребру 16, вес общего пути будет 0. Поэтому выгоднее идти через вершину 3.

Во втором тесте мы можем пройти либо по цепочке $1 \rightarrow 2 \rightarrow 3 \rightarrow 6$, либо $1 \rightarrow 4 \rightarrow 5 \rightarrow 6$. В первом случае вес пути будет $7 \text{ AND } 8 \text{ AND } 7 = 0$, во втором случае вес пути будет $5 \text{ AND } 9 \text{ AND } 9 = 1$.

В третьем тесте нет ни одного пути из вершины 2 в вершину 1.

Задача J. Задача о Серёже

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

Серёже сегодня пять лет! Когда ему исполнился год, родители подарили ему число, когда ему исполнилось два годика, родители подарили ему массив целых чисел. На трёхлетие он получил строку. Когда он подрос до четырёх, его разбудила мама, тихим голосом пожелала ему быть хорошим мальчиком и дала подвешенное дерево. Сегодня он стал совсем взрослым. Он нашёл ориентированный граф без петель, подаренный ему родителями.

Так как Серёжа очень любознательный мальчик, он сразу придумал себе занятие. Он решил найти в этом графе такое множество вершин Q , что никакие две вершины $x, y \in Q$ не соединены ребром, а также до любой вершины $z \notin Q$ можно было дойти от какой-нибудь вершины множества Q не более, чем за два шага.

Немного подумав, Сережа смог решить поставленную задачу, он ведь уже такой большой. А сможете ли вы?

Формат входных данных

Первая строка содержит два положительных целых числа n и m ($1 \leq n \leq 1\,000\,000$, $1 \leq m \leq 1\,000\,000$) — количество вершин и рёбер в ориентированном графе.

Каждая из следующих m строк содержит описание очередного ребра. Каждая из этих строк содержит два числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), задающие начало и конец i -го ребра. Граф может содержать кратные рёбра.

Формат выходных данных

В первой строке выведите число k — количество выбранных вершин. А во второй строке выведите эти k чисел — номера выбранных вершин.

Если существует несколько подходящих множеств — выведите любое из них. В частности, не требуется находить множество минимального размера. Гарантируется, что хотя бы одно подходящее множество всегда существует.

Примеры

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

Замечание

В первом примере вершины 1, 3, 4, 5 не соединены. До вершины 2 можно дойти за один шаг от вершины 1.

Во втором примере до вершины 1 можно дойти за один шаг, а до вершины 2 за два шага.