

Задача А. Смешарики: Быстрое преобразование Фурье

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

Кар-Карыч купил Лосяшу билеты на премьеру очередного фильма про Смешариков «Смешарики: Быстрое преобразование Фурье». Места для зрителей в зале кинотеатра, в котором будет показываться премьера, расположены в виде прямоугольника. Место определяется номером ряда (по вертикали, сверху вниз) и номером места в ряду (по горизонтали, слева направо).

Кар-Карыч знает, что Лосяш любит думать про числа, поэтому хочет зашифровать его место в кинотеатре. Для этого он пронумеровал все места по очереди, сверху вниз и слева направо, сначала нумеруя все места первого ряда слева направо, затем, продолжая нумерацию, все места второго ряда слева направо и так далее до последнего ряда.

Кар-Карыч хочет сказать Лосяшу его место в такой нумерации, но он не очень силен в математике. Помогите ему!

Формат входных данных

В единственной строке входных данных содержится четыре целых числа h , w , x и y ($1 \leq h, w \leq 10^9$, $1 \leq x \leq h$, $1 \leq y \leq w$) — количество рядов в зале кинотеатра, количество мест в каждом ряду, номер ряда и номер места в ряду соответственно.

Формат выходных данных

Выведите одно число — номер места в нумерации Кар-Карыча.

Пример

стандартный ввод	стандартный вывод
3 4 2 3	7

Замечание

Для первого тестового случая нумерация сидений выглядит следующим образом, нужное место выделено серым:

Номер кресла →	1	2	3	4
Ряд ↓				
1	1	2	3	4
2	5	6	7	8
3	9	10	11	12

Задача В. Время суток

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

Определите время суток по времени на часах.

Чтобы проверить правильность вашего ответа, мы провели социологический опрос и выяснили, что утро среднестатистического человека длится с 06:36 до 11:46, день начинается в 11:47 и заканчивается в 17:58, вечер начинается в 17:59, сразу же после конца дня, и продолжается до 00:42, а всё остальное время идёт ночь.

Возможно, для вас границы между временами суток расположены по-другому, однако от вас потребуются ответы, наиболее точные для большинства людей, то есть согласованные с нашими результатами.

Формат входных данных

В единственной строке вводится строка в формате `hh:mm` — время на часах от 00:00 до 23:59.

Формат выходных данных

Выведите `Morning`, `Day`, `Evening` или `Night` — время суток, соответствующее времени на часах.

Примеры

стандартный ввод	стандартный вывод
11:30	Morning
00:42	Evening

Замечание

Не допускайте опечаток в выводе.

Задача С. Авантюра

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

Стёпа после пройденного на «Степике» курса по машинному обучению решил написать нейросеть, которая определяет *корректность* номера банковской карты, но, так как Стёпа делает только первые шаги в написании нейросетей, для её обучения он смог найти лишь много файлов, в каждом из которых лежит по четыре строки s_i .

Четыре строки образуют *корректный* номер банковской карты, если:

1. Каждая строка состоит из четырёх цифр;
2. В каждой строке чётность цифр совпадает с чётностью позиции, если считать, что старший разряд имеет номер 1. Другими словами, строка должна выглядеть как нечётное – чётное – нечётное – чётное.

Ваша задача, зная алгоритм проверки корректности номера банковской карты, помочь Стёпе определить, в каком файле правильный номер карты, а в каком нет, чтобы Стёпа по этим данным смог обучить нейросеть.

Формат входных данных

На вход через пробел подаются четыре строки s_i ($1 \leq |s_i| \leq 9$), состоящих только из цифр 0 – 9.

Формат выходных данных

Выведите **Yes**, если эти четыре строки являются корректным номером банковской карты, иначе **No**.

Примеры

стандартный ввод	стандартный вывод
1010 1010 1010 1010	Yes
1111 1111 1111 1111	No
00 12345 9 111111	No
1234 5678 9012 3456	Yes

Замечание

В первом примере все строки удовлетворяют обоим пунктам, поэтому номер корректный.

Во втором примере на чётных позициях стоят нечётные цифры, поэтому номер некорректный.

Задача D. Гнездовские игры

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

2024 год объявлен президентом Российской Федерации годом семьи. Услышав эту новость, Петя, только что закончивший исторический факультет ЧелГУ, решил, что настало время вернуться в свою родную деревню Гнездо. Через два часа он, очень радостный, уже сидел за столом в отцовской избушке и, не торопясь, поедал мамин борщ.

В деревне было скучно жить. Петя перепробовал все забавы, которые только мог. Через месяц он уже знал, что в деревне проживает ровно N семей, которые имеют номера от 1 до N . В каждой семье может быть сколько угодно человек, однако Петя не обладает хорошей внимательностью, поэтому в каких-то семьях может не быть ни одного.

Исторический факультет Челябинского государственного университета даёт человеку весьма обширный кругозор, улучшает память и работу мозга в целом. Поэтому одним холодным октябрьским вечером Петя решил вспомнить все даты, которые когда-то учил. Проведя немного времени за таким увлекательным занятием, он понял, что ему обязательно нужно войти в историю своей родной деревни. Петя решил, что должен провести «Гнездовские игры» — спортивное соревнование среди всех детей деревни. После чего он сразу же оповестил всех о начале регистрации на такое грандиозное событие.

Уже через час у Пети был список всех детей деревни, упорядоченный по дате рождения детей. Для каждого ребёнка известен номер его семьи и пол. Более формально, у Пети есть список из K строк, каждая из которых содержит число A_i (номер семьи) и символ B_i (М для мальчика, F для девочки) — описание i -го в порядке рождения ребёнка.

Сначала Петя понял, что информация о девочках ему совсем не понадобится, так как соревнования будут по боксу. В деревне Гнездо не поддерживают драки между лицами женского пола. После этого Петя понял, что на соревнования зарегистрировалось слишком много детей. Поэтому от каждой семьи он хочет оставить только нескольких представителей.

Брат самого старшего мальчика от каждой семьи Пете слишком скучно. Брат самого младшего — слишком опрометчиво, так как тогда некоторым участникам придётся сражаться в подгузниках, что запрещено правилами. Поэтому Петя решил взять от каждой семьи **среднего по возрасту мальчика**. Если таких в семье двое, то в соревнованиях будут участвовать оба.

Петя знает, что он очень невнимательный, поэтому просит вас о помощи. Определите для каждого ребёнка, будет ли он участвовать в «Гнездовских играх».

Уточним, что «**средний по возрасту мальчик**» в семье — это тот мальчик, который будет находиться ровно посередине списка детей этой семьи, упорядоченных по дате рождения, из которого удалили всех девочек. Если в семье чётное ненулевое количество мальчиков, то «средними по возрасту» будут двое.

Формат входных данных

Первая строка содержит два целых числа N и K ($1 \leq N, K \leq 100$) — количество семей и детей соответственно.

i -я из следующих K строк содержит описание i -го ребенка — целое число A_i ($1 \leq A_i \leq N$) и один символ B_i (либо M, либо F).

Формат выходных данных

Выведите K строк.

В i -й строке выведите Yes, если i -й ребёнок будет участвовать в «Гнездовских играх», и No иначе.

Примеры

стандартный ввод	стандартный вывод
3 5	No
3 M	Yes
1 M	Yes
3 M	No
3 M	No
2 F	
1 5	No
1 M	Yes
1 M	Yes
1 M	No
1 F	No
1 M	

Замечание

Разберём первый пример из условия.

- В семье с номером 1 ровно один ребёнок, и это мальчик. Он будет участвовать.
- В семье с номером 2 ровно один ребёнок, и это девочка. Тут никто не участвует.
- В семье с номером 3 ровно три ребёнка, и каждый из них мальчик. Средний из них будет участвовать.

Разберём второй пример из условия.

- В семье с номером 1 ровно пять детей, и четыре из них мальчики. В этой семье двое мальчиков будут средними — 2-й и 3-й — они оба будут участвовать.

Задача Е. Командировки

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

В Берляндии используется необычный календарь. В году 10^{10} дней, и нет никаких месяцев — дни просто нумеруются по порядку натуральными числами от 1 до 10^{10} .

Поликарпу, как самому ответственному сотруднику ООО «Галера», руководство назначило n командировок. i -я командировка начинается в день с номером a_i и длится 7 дней, то есть Поликарп находится в командировке в дни с номерами от a_i до $(a_i + 6)$ включительно. Командировки, разумеется, не пересекаются, ведь Поликарп не может быть в двух местах одновременно.

В конце года настала пора выплачивать командировочные, но в бухгалтерии, как обычно, всё перепутали и потеряли часть данных о командировках Поликарпа. Чтобы правильно рассчитать командировочные, бухгалтерии необходимо узнать информацию о q днях — про каждый из них необходимо выяснить, был ли в этот день Поликарп в командировке или нет. Помогите им с этим.

Формат входных данных

Первая строка входных данных содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество командировок.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — номера дней начала командировок.

В третьей строке содержится число q ($1 \leq q \leq 2 \cdot 10^5$) — количество дней, информацию о которых хочет узнать бухгалтерия.

В последней строке содержится q целых чисел b_1, b_2, \dots, b_q ($1 \leq b_i \leq 10^9$) — номера дней, информацию о которых нужно узнать бухгалтерии.

Формат выходных данных

Вы должны вывести q строк. В i -й строке выведите **Yes**, если Поликарп был в командировке в день с номером b_i , и **No** иначе.

Пример

стандартный ввод	стандартный вывод
2	Yes
31 42	No
7	Yes
36 58 43 115 42 31 32	No
	Yes
	Yes
	Yes

Задача F. Такси

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

Полицейский Эмильен пытается поймать неуловимую банду на красных мерседесах. Различные облавы и засады ни к чему не приводят. Отчаявшись, Эмильен обращается за помощью к своему другу Даниэлю. Даниэль ничего не смыслит в следственных мероприятиях, да и вообще недолюбливает представителей правопорядка, но отказать другу не в силах. Друзья отправляются на место последнего ограбления, где Даниэль обнаруживает отпечаток рисунка протектора автомобиля преступников. Теперь, чтобы найти нарушителей закона, необходимо выяснить где они покупают колеса для своих машин. По счастливой случайности в полицейском участке есть база автомобильных покрышек, продающихся во Франции. Программисты в участке уже написали алгоритм перебора фотографий покрышек, но еще не реализовали сравнение отпечатка рисунка протектора на земле с самим протектором, потому что у них закончился рабочий день. Эмильен и Даниэль просят вас дописать программу и помочь им поймать преступников.

Фотография отпечатка совпадает с фотографией покрышки, если фотографию покрышки можно сколько-то раз повернуть (**не переворачивать**), чтобы фотографии *совпали по трафарету*.

Две фотографии *совпали по трафарету*, если не существует клетки (i, j) , в которой у фотографий стоят одинаковые символы.

Формат входных данных

В первой строке вводится целое число N ($1 \leq N \leq 100$) — размер фотографий, используемых в участке.

Далее вводится фотография отпечатка: N строк длины N , содержащих только символы «#» и «_».

После чего следует пустая строка и фотография покрышки: N строк длины N из символов «#» и «_».

Формат выходных данных

В качестве ответа выведете слово **Yes**, если рисунок протектора на покрышке соответствует отпечатку, и **No** в противном случае.

Примеры

стандартный ввод	стандартный вывод
5 ##### #_#_# ##_## #_#_# ##### ----- _#_#_ _###_ _#_#_ -----	Yes
3 ### ### --- --- _#_ ###	No

Замечание

В первом тесте одну из фотографий повернуть на 90 градусов, и рисунки совпадут полностью.
Во втором тесте рисунки не совпадут ни при каких условиях.

Задача G. Громкость и три бита

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

После окончания ЧеРеКОШа Данил решил посмотреть милые видео про котят, однако обнаружилось, что громкость телефона выставлена на 0. Данил захотел увеличить громкость до n , поэтому вспомнил, что умеет увеличивать громкость на любое целое положительное число x , такое что двоичная запись числа x имеет ровно три единичных бита. Он может увеличивать громкость таким образом любое количество раз (даже ноль).

Помогите Данилу определить, сможет ли он после какого-то количества операций прибавления громкости получить громкость на телефоне ровно n ?

Напомним, что число состоит из трёх единичных битов, если двоичная запись этого числа имеет ровно три единицы.

Формат входных данных

В первой строке содержится целое число t ($1 \leq t \leq 10\,000$) — количество наборов входных данных.

В следующих t строках содержится по одному целому числу n ($0 \leq n \leq 10^{18}$) — желаемая громкость для очередного набора данных.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите **Yes**, если можно описанными операциями получить желаемую громкость, или **No**, если нельзя.

Ответ можно выводить в любом регистре.

Пример

стандартный ввод	стандартный вывод
3	No
3	Yes
28	No
23	

Замечание

28 можно получить следующим образом: $28 = 7 + 7 + 14$.

Задача Н. Хорошая задача с хорошим условием про арбузы

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

В большой комнате в ряд стоят N детей. Каждый в руках держит арбуз. i -й ребёнок держит арбуз веса a_i (удивительно, но вес арбуза может быть 0).

В комнату заходит Фокусник, подходит к некоторому ребёнку и достаёт у него из-за правого уха конфетку. Конечно, всем детям, стоящим справа от этого ребёнка, становится интересно, что же там такое происходит. Но, к сожалению, пытаясь увидеть фокус, эти дети теряют концентрацию и с грохотом роняют свои арбузы (хотя возможно, что Фокусник выберет такого ребёнка, что никто не уронит арбуз). Потом Фокусник снова подходит к некоторому ребёнку, у которого остался арбуз (возможно, к тому же самому), но теперь достаёт конфету из-за левого уха. И снова дети роняют свои арбузы. Только теперь роняют те, которые стоят *слева* от счастливого, у которого за ухом обнаруживается конфета.

Фокусник на этом прекращает фокус с конфетами, но зато в комнату ввозят огромные весы, и дети выясняют суммарный вес уцелевших арбузов. Если суммарный вес оказывается равен волшебному числу K , то Фокусник легко сможет превратить все разбитые арбузы в целые. Иначе разбитые арбузы остаются разбитыми, и дети делят между собой лишь уцелевшие арбузы.

Вам, как страшному арбузоненавистнику, интересно, сколько существует способов Фокуснику выбрать ребят, у которых он будет доставать конфету из-за уха, так, чтобы у Фокусника не было возможности восстановить арбузы (два способа считаются различными, если различны наборы детей, оставшихся с арбузами в руках).

Формат входных данных

В первой строке даны два целых числа N и K ($1 \leq N \leq 10^5$, $0 \leq K \leq 10^9$) — количество детей и волшебное число.

Вторая строка содержит N целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — веса арбузов.

Формат выходных данных

Выведите единственное число — количество способов выбрать двух ребят для фокуса, чтобы у Фокусника не было возможности восстановить разбитые арбузы.

Примеры

стандартный ввод	стандартный вывод
4 2 1 1 1 1	7
3 300 0 3 1	6

Замечание

Первый тест: если арбузы останутся у детей с номерами (в 1-индексации) 1 и 2 или 2 и 3 или 3 и 4, то Фокусник сможет восстановить арбузы. При любом другом наборе детей — не сможет. Таких других наборов 7 штук.

Второй тест: кого бы ни выбирал Фокусник, он никогда не сможет восстановить арбузы.

Задача I. Гирлянда из флажков

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

Близится день рождения Совуны, и Нюша решила подарить ей гирлянду из флажков. Гирлянды из флажков были популярны в молодости Совуны, поэтому остался всего лишь один магазин, где они продаются, куда и пошла Нюша.

Гирлянды в том магазине нужно собирать из продающихся n флажков. Все флажки имеют разный дизайн, но на каждом флажке написана маленькая буква английского алфавита. Флажки выложены в ряд, на i -м флажке написана буква s_i , и стоит он c_i монет. Каждый из n флажков можно купить не больше одного раза.

Нюша хочет собрать красивую гирлянду, поэтому ей хочется купить флажки такие, чтобы из них можно было собрать гирлянду, буквы на которой составляют строку-палиндром. Палиндром — это строка, которая читается одинаково как слева направо, так и справа налево. Например, строки `abacaba`, `aaaa`, `abba`, `racacar` — палиндромы.

У Нюши есть всего w монет, которые она может потратить на подарок, но она хочет сделать как можно более большой подарок, поэтому она хочет собрать гирлянду из наибольшего возможного количества флажков так, что буквы на ней составляют палиндром и суммарная стоимость всех флажков, вошедших в гирлянду, не превышает w .

Помогите ей и скажите максимальную длину такой гирлянды и номера флажков, которые ей нужно купить!

Формат входных данных

Первая строка входных данных содержит два целых числа n и w ($1 \leq n \leq 10^5$, $1 \leq w \leq 10^9$) — количество флажков в магазине и бюджет Нюши в монетах соответственно.

Вторая строка входных данных содержит строку s длины n , состоящую из маленьких букв английского алфавита, где s_i — буква, написанная на i -м флажке.

Третья строка входных данных содержит массив натуральных чисел c длины n , где c_i — стоимость i -го флажка в монетах ($1 \leq c_i \leq 10^9$).

Формат выходных данных

В первой строке выходных данных выведите одно число k — максимальную длину гирлянды, буквы на которой составляют палиндром и которая укладывается в бюджет Нюши.

Во второй строке выходных данных выведите k различных натуральных чисел, разделенных пробелом — номера флажков, которые Нюше нужно купить. Номера можно выводить в любом порядке.

Если существует несколько вариантов купить флажки так, чтобы получить гирлянду-палиндром максимальной длины, можно вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
5 10 abcce 5 7 2 2 2	3 3 4 5
3 1 aaa 3 1 2	1 2
4 23 aabb 3 17 10 10	3 1 3 4

Замечание

В первом примере, если взять флажки с номерами 3, 4 и 5, нужно будет заплатить $2 + 3 + 2 = 7$ монет, что меньше 10, и составить палиндром *ses*. Также можно взять флажки с номерами 1, 3 и 4 и составить палиндром *sas*, заплатив 10 монет.

Задача J. Тимофей и подготовка к ЧеРеКОШу

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

Мальчик Тимофей очень хочет подготовиться к ЧеРеКОШу. Он готов на всё ради этого, готов даже пожертвовать своим летним отдыхом. Поэтому он всё лето решал задачи на разных сайтах, в том числе на сайте CodeForceless, на котором ведётся подсчёт решённых задач.

Но так как у сайта CodeForceless слишком слабые сервера, а Тимофей решил невероятно много задач, сайт упал, и теперь к нему нет доступа. Тимофей хочет похвастаться перед друзьями результатами подготовки и рассказать им, сколько задач он решил за лето, но само число вылетело у него из головы. Единственное, что он запомнил, это количество вхождений каждой цифры, а также что никакие две одинаковые цифры в числе не стоят рядом.

Чтобы не показаться хвастуном и не преувеличить, Тимофей хочет предоставить минимальное число с таким свойством или же понять, что такого числа не существует. Помогите молодому олимпиаднику с этой задачей.

Формат входных данных

Входные данные состоят из 10 строк.

i -я строка содержит число a_i ($0 \leq a_i \leq 2 \times 10^5$) — количество цифр $(i-1)$ в числе задач, решённых

Тимофеем. $1 \leq \sum_{i=1}^n a_i \leq 2 \times 10^5$.

Формат выходных данных

Если такое целое неотрицательное число существует, выведите его в единственной строке, иначе — строку Impossible.

Примеры

стандартный ввод	стандартный вывод
0 0 0 0 0 0 0 0 0 0 2	Impossible
1 3 0 0 5 4 0 0 0 0	1014145454545

Замечание

В числе не может быть ведущих нулей (только если оно не равно нулю).

Задача К. Спасти любой ценой

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

Окабэ Ринтаро в отчаянии. Его ассистентка и подруга Курису Макисэ вчера была убита при захвате их лаборатории. Неожиданно в его комнату врывается Судзуха Аманэ, представляется путешественницей во времени и сообщает, что Курису осталась жива ровно в одной временной линии, да и в ней она была захвачена сотрудниками *Организации*. Наш герой во что бы то ни стало решает спасти её.

Судзуха сообщает Окабэ, что они сейчас находятся во временной линии со значением *дивергенции* равным s , а его ассистентка — со значением t . Проанализировав n временных веток, Ринтаро придумал m «D-mail'ов», посылая которые себе в прошлое, можно перемещаться между временными линиями. Для удобства он принял текущее время равным 0 и установил, что i -й «D-mail» можно отправить, находясь только в ветке с *дивергенцией* равной u_i . Пусть в этой ветке текущее время равно t_0 , тогда после отправки «D-mail'а» герой переместится в ветку с *дивергенцией* равной v_i , а время в ней будет равно $t_0 + w_i$. Чтобы не рвать ткань пространственно-временного континуума окончательно, Ринтаро решил не отправлять следующее сообщение, пока предыдущее не будет прочитано и, соответственно, он не переместится в другую ветку.

Однако не всё так просто. Чтобы сообщение подействовало, необходимо, чтобы Ринтаро из прошлого его прочитал, однако это произойдёт, только если его телефон заряжен. Окабэ из прошлого, которому придёт i -е сообщение, в очередной раз включает свой телефон в d_i секунду, тот держит заряд ровно g_i секунд, после чего выключается, и герой заряжает его следующие p_i секунд, лишь после чего включает телефон и может вновь увидеть сообщение. Этот процесс повторяется бесконечно. Более формально, телефон включен \dots , с $d_i - g_i - p_i$ по $d_i - p_i - 1$, с d_i по $d_i + g_i - 1$, с $d_i + g_i + p_i$ по $d_i + 2 \cdot g_i + p_i - 1$, \dots секунды включительно.

Но и это ещё не всё. Из-за безответственных игр со временем во временных линиях появились аномалии, по одной в каждой линии: в i -й аномалия начнётся в l_i секунду и продлится вплоть до r_i секунды включительно. Попасть в любую такую смерти подобно, что для него, что для Курису, поэтому они должны избегать их.

План Окабэ заключается в следующем: он перемещается в одну ветку с ещё живой Курису, освобождает её (мгновенно, его гаджеты помогут ему в этом), и они вместе перемещаются в уже *безопасную* ветку, т.е. в такую, в которой аномалия уже закончилась. На этом операция по спасению оканчивается. К сожалению, далеко не факт, что операция пройдёт так гладко. Возможны и три других исхода: Окабэ освободит Курису, но они попадут вдвоём в аномалию. Окабэ не успеет освободить Курису, и она попадёт в аномалию, но сам он сможет добраться до *безопасной* ветки. И, наконец, худший: Окабэ ни Курису не освободит, ни сам не избежит попадания в аномалию. Конечно, Окабэ предпочтёт попасть в аномалию вместе с подругой, нежели сбежать и выжить, предав её, однако если он и добраться до неё не успеет, то предпочтёт спастись самому. Более того, если ему(им) придётся расстаться с жизнью, то он(и) желает(ют), чтобы это случилось как можно позже, и чтобы в момент попадания в аномалию он(и) успели окинуть взглядом этот мир хоть на секунду. Иными словами, аномалия должна застичь героя(ев) не ранее, чем через секунду, после последнего перемещения (если таковое было), и среди всех таких вариантов он(и) выберут тот, в котором это случится как можно позже).

Окабэ просит помощи с подсчётами у Аманэ, но не знает, что то же самое попросили ещё несколько его копий из параллельных вселенных, всего T . У Судзухи уже голова идёт кругом от всего этого, помогите ей с вычислениями.

Формат входных данных

В первой строке записано одно целое число T ($1 \leq T \leq 10^5$) — количество параллельных вселенных, в которых Окабэ требуется помощь. Далее следует описание каждой.

В первой строке описания каждой вселенной содержится два целых числа n и m ($2 \leq n \leq 10^5$,

$0 \leq m \leq 10^5$) — количество временных линий и количество придуманных Окабэ «D-mail'ов» соответственно.

В i -й из следующих m строк даны шесть чисел $u_i, v_i, w_i, d_i, g_i, p_i$ ($1 \leq u_i, v_i \leq n$, $1 \leq w_i, g_i, p_i \leq 10^9$, $0 \leq d_i < g_i + p_i$) — значения *дивергенции* временных линий, в которой нужно находиться, чтобы отправить i -е сообщение, и той, в которую оно перемещает, разница во времени при перемещении, момент времени, в который адресат этого письма в очередной раз включит телефон, сколько последний держит заряд и сколько его нужно заряжать соответственно.

В следующих n строках дано описание аномалий. В i -й из них даны два целых числа l_i, r_i ($1 \leq l_i \leq r_i \leq 10^9$) — время начала и конца аномалии во временной линии с *дивергенцией* равной i .

Наконец, в последней строке описания вселенной даны два целых числа s и t ($1 \leq s, t \leq n$, $s \neq t$) — *дивергенции* временных линий, в которых изначально находятся Окабэ и Курису соответственно.

Гарантируется, что сумма n и m по всем параллельным вселенным, в которых Окабэ требуется помощь, не превосходит 10^5 .

Формат выходных данных

Если операция по спасению может завершиться удачно, выведите **Happy end in t sec**, где t — минимально возможное время в секундах, в которое может завершиться операция.

Если Окабэ успеет освободить Курису, но они не смогут избежать смерти, выведите **Tragic death in t sec**, где t — максимально возможное время попадания героев в аномалию при соблюдении всех описанных условий.

Если Окабэ не успеет спасти Курису, но сможет добраться до безопасной ветки сам, выведите **Traitor's escape in t sec**, где t — минимально возможное время в секундах, в которое Окабэ может оказаться в уже *безопасной* ветке.

Наконец, если Окабэ и Курису суждено попасть в аномалию по отдельности, выведите **Loser's death in t sec**, где t — максимально возможное время попадания Окабэ в аномалию при соблюдении всех описанных условий.

Пример

стандартный ввод	стандартный вывод
4	Traitor's escape in 5 sec
2 2	Tragic death in 6 sec
1 2 5 0 1 2	Loser's death in 1 sec
2 2 10 1 1 1	Happy end in 4 sec
1 2	
4 4	
1 2	
2 2	
1 2 5 0 1 2	
1 1 10 0 1 1	
1 2	
6 6	
1 2	
2 2	
1 2 5 0 1 2	
2 2 10 0 1 1	
1 2	
5 5	
1 2	
2 2	
1 2 2 0 1 1	
2 1 2 0 1 1	
1 3	
3 10	
1 2	

Замечание

Разные временные ветки имеют разные значения *дивергенции*.

Сообщения отправляются и прочитываются мгновенно.

В первой параллельной вселенной Окабэ в нулевую секунду должен отправить первый «D-mail» и переместиться в линию с Курису за 5 секунд, однако та к этому моменту уже будет мертва, поэтому спасти он её не успеет, при этом он может остаться в этой линии, т.к. она уже *безопасна*.

Во второй параллельной вселенной Окабэ, отправив первый «D-mail» в нулевую секунду, успеет освободить Курису, но больше переместится они никуда не смогут, поэтому смерть их настигнет в 6 секунду. Заметим, что Окабэ также мог оставаться в исходной временной линии сколь угодно долго, отправив второй «D-mail» в нулевую секунду, однако это для него менее предпочтительно.

В третьей вселенной Окабэ не успеет добраться до Курису до её смерти и не сможет добраться до уже *безопасной* линии, поэтому умрёт через секунду, после начала операции в исходной временной линии. Отметим, что он не станет перемещаться, т.к. иначе умрёт сразу же после перемещения, а не хотя бы через секунду, как он того хочет.

В четвёртой вселенной Окабэ отправляет первый «D-mail» в нулевую секунду, спасает Курису во вторую и сразу же отправляет второй «D-mail». Таким образом, они вдвоём оказываются в исходной временной линии в 4 секунду, а она к тому моменту уже станет *безопасной*.

Задача L. Яблоне́вый сад

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Мальчик Женя захотел посадить в саду яблоню. Однако его жизнь была достаточно необычной, поэтому в своём возрасте из всех деревьев он знал только двоичные деревья поиска.

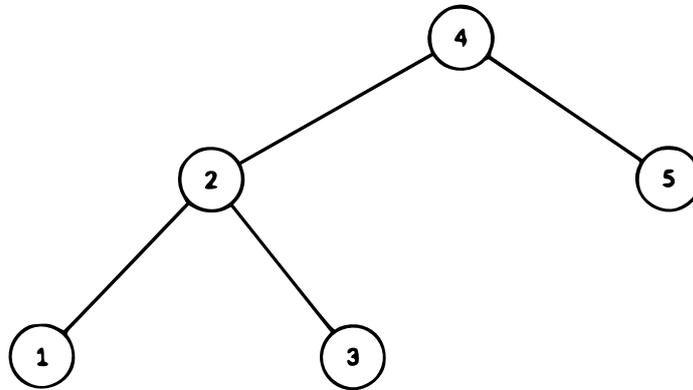
Двоичное дерево — это корневое дерево, в котором у каждой вершины есть не более двух сыновей (их называют *левым* и *правым* сыном).

Двоичное дерево поиска — это двоичное дерево, в каждой вершине которого записано число k , называемое *ключом вершины*, при этом верно следующее свойство:

- Для любой вершины u из поддерева левого сына v выполняется $k_u < k_v$;
- Для любой вершины u из поддерева правого сына v выполняется $k_v < k_u$.

Женя придумал двоичное дерево поиска, ключами в котором является перестановка размера n . Однако сегодняшний вечер у Жени уже был занят: он планировал встретиться с друзьями в неизвестном ресторане «Брёвна». Поэтому, чтобы не забыть дерево, Женя решил записать информацию о нём на листочек. Рисовать он не умел, поэтому придумал специальный метод шифрования, чтобы гарантированно вспомнить дерево после встречи с друзьями:

```
def encode_tree(v):  
    if v is None:  
        return []  
    return [v.x] + encode_tree(v.left) + encode_tree(v.right)
```



Для приведённого дерева результат функции `encode_tree` будет `[4, 2, 1, 3, 5]`.

Однако Женя перехитрил сам себя: вернувшись домой вечером, он понял, что никак не может расшифровать написанные на листочке числа. Создание яблоневого сада под угрозой, поэтому вам требуется помочь Жене по заданной зашифрованной записи восстановить исходное дерево, либо же сообщить, что Женя что-то напутал, и по зашифрованной записи восстановить дерево невозможно.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 300\,000$) — количество вершин в дереве.

Во второй строке задано n целых чисел k_i ($1 \leq k_i \leq n$) — результат выполнения функции `encode_tree`. Гарантируется, что массив k_i — перестановка чисел от 1 до n .

Формат выходных данных

Если исходные данные противоречивы, и восстановить дерево невозможно, то выведите `Impossible`.

В противном случае выведите n строк. В i -й строке выведите два числа: ключи левого и правого сына вершины с ключом i . Если левый или правый сын отсутствует, то выведите -1 .

Примеры

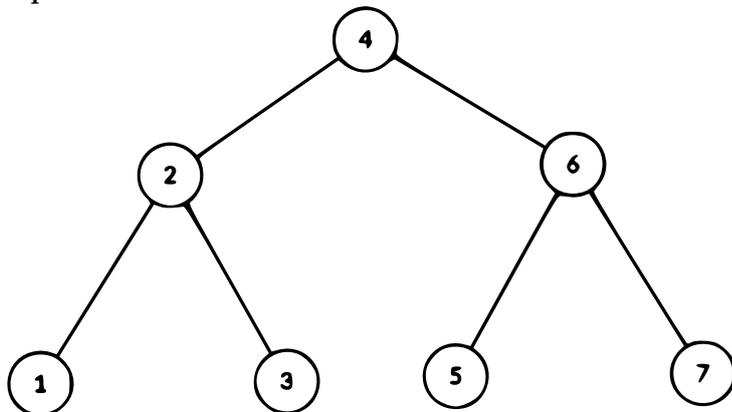
стандартный ввод	стандартный вывод
5 4 2 1 3 5	-1 -1 1 3 -1 -1 2 5 -1 -1
3 2 3 1	Impossible
7 4 2 1 3 6 5 7	-1 -1 1 3 -1 -1 2 6 -1 -1 5 7 -1 -1
3 1 2 3	-1 2 -1 3 -1 -1

Замечание

Первый тест — это пример из условия.

Во втором тесте подходящего дерева не существует.

Третий тест:



Четвёртый тест:

