

Задача А. Зажигаем светлячков

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

Вдоль сцены расположены N фонарей, пронумерованных слева направо числами от 1 до N .

Изначально на фонаре с номером i сидит светлячок с яркостью A_i .

Перед началом ночного представления режиссёр может один раз изменить расположение светлячков. Он должен рассадить их в любом порядке так, чтобы на каждом фонаре оказался ровно один светлячок.

Если светлячок, который изначально сидел на фонаре с номером x , после перестановки окажется на фонаре с номером y , то он принесёт представлению

$$A_x \cdot |x - y|$$

очков зрелищности.

Найдите максимальную суммарную зрелищность, которую можно получить, оптимально переставив светлячков.

Формат входных данных

В первой строке дано целое число N — количество фонарей и светлячков ($2 \leq N \leq 2000$).

Во второй строке даны N целых чисел $A_1, A_2, \dots, A_N$ — яркости светлячков ($1 \leq A_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — максимальную суммарную зрелищность представления.

Примеры

стандартный ввод	стандартный вывод
4 1 3 4 2	20
6 5 5 6 1 1 1	58
6 8 6 9 1 2 1	85

Замечание

В первом примере можно выполнить следующую перестановку:

- светлячка с фонаря 1 пересадить на фонарь 3;
- светлячка с фонаря 2 пересадить на фонарь 4;
- светлячка с фонаря 3 пересадить на фонарь 1;
- светлячка с фонаря 4 пересадить на фонарь 2.

Тогда суммарная зрелищность будет равна

$$1 \cdot |1 - 3| + 3 \cdot |2 - 4| + 4 \cdot |3 - 1| + 2 \cdot |4 - 2| = 20.$$

Получить больше 20 очков невозможно.

Задача В. Распространение гномиков

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

Лондонское метро представляет собой n станций, пронумерованных от 1 до n . Станции соединены между собой $n - 1$ двусторонним тоннелем. Тоннель номер i соединяет между собой станции v_i и u_i . По тоннелям можно добраться от любой станции до любой другой.

Шерлок Гномс может отправить своих приспешников на не более чем k станций. Основной задачей гномиков-приспешников является распространение информации.

В момент времени 0 информация считается распространенной на тех станциях, куда Шерлок Гномс отправил своих приспешников изначально. Далее за единицу времени информация распространяется от станции ко всем ее соседям.

Шерлок Гномс хочет выбрать, на какие k станций отправлять своих приспешников, так, чтобы время, в которое информация распространилась на последней станции, было минимальным. Помогите ему с этим.

Формат входных данных

В первой строке даны два целых числа n и k ($1 \leq k < n \leq 2 \cdot 10^5$) – количество станций в Лондонском метро и количество станций, на которые Шерлок Гномс может отправить своих приспешников.

В каждой из следующих $n - 1$ строк даны два целых числа u и v ($1 \leq u, v \leq n$) – станции, соединённые очередным тоннелем.

Гарантируется, что по тоннелям можно добраться от любой станции до любой другой.

Формат выходных данных

Выведите минимальное время, за которое информация может распространиться по всем станциям.

Примеры

стандартный ввод	стандартный вывод
5 2 1 2 2 3 3 4 4 5	1
5 1 1 2 1 3 1 4 5 4	2
20 3 2 15 6 5 12 1 7 9 17 2 15 5 2 4 17 16 12 2 8 17 17 19 18 11 20 8 20 3 13 9 11 10 11 20 14 8 11 7	3

Замечание

В первом примере, если Шерлок Гномс отправит своих приспешников на станции 2 и 4, информация распространится по станциям 1, 2, 3, 4, 5 в моменты времени 1, 0, 1, 0, 1 соответственно. То есть по всем станциям информация распространится к моменту времени 1. Можно показать, что за меньшее время информация не может распространиться по всем станциям.

Задача С. Роботы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Младший брат Мариты разбросал игрушки по всему полу в гостиной. К счастью, Марита разработала специальных роботов, которые могут убрать игрушки. Ей нужна ваша помощь, чтобы определить, какие роботы должны убирать какие игрушки.

В комнате находятся T игрушек. Игрушка i имеет целочисленный вес W_i и целочисленный размер S_i .

Существуют роботы двух типов: слабые и маленькие.

- Есть A слабых роботов. Слабый робот i имеет ограничение по весу X_i и может убрать любую игрушку, вес которой строго меньше X_i . Размер игрушки для такого робота не имеет значения.
- Есть B маленьких роботов. Маленький робот i имеет ограничение по размеру Y_i и может убрать любую игрушку, размер которой строго меньше Y_i . Вес игрушки для такого робота не имеет значения.

Каждому роботу требуется одна минута, чтобы убрать одну игрушку. Разные роботы могут одновременно убирать разные игрушки.

Определите, могут ли роботы Мариты убрать все игрушки. Если это возможно, найдите минимальное количество минут, необходимое для этого.

Формат входных данных

В первой строке содержатся три целых числа A , B и T — количество слабых роботов, количество маленьких роботов и количество игрушек соответственно.

Во второй строке содержатся A целых чисел $X_1, X_2, \dots, X_A$ — ограничения по весу для слабых роботов.

Если $A = 0$, вторая строка является пустой.

В третьей строке содержатся B целых чисел $Y_1, Y_2, \dots, Y_B$ — ограничения по размеру для маленьких роботов.

Если $B = 0$, третья строка является пустой.

В следующих T строках содержатся по два целых числа W_i и S_i — вес и размер игрушки i .

Гарантируется, что выполняются следующие ограничения:

- $0 \leq A \leq 50\,000$;
- $0 \leq B \leq 50\,000$;
- $1 \leq A + B$;
- $1 \leq T \leq 1\,000\,000$;
- $1 \leq X_i \leq 2\,000\,000\,000$ для всех $1 \leq i \leq A$;
- $1 \leq Y_i \leq 2\,000\,000\,000$ для всех $1 \leq i \leq B$;
- $1 \leq W_i \leq 2\,000\,000\,000$ для всех $1 \leq i \leq T$;
- $1 \leq S_i \leq 2\,000\,000\,000$ для всех $1 \leq i \leq T$.

Формат выходных данных

Если убрать все игрушки невозможно, выведите -1 .

Иначе выведите одно целое число — минимальное количество минут, за которое роботы могут убрать все игрушки.

Примеры

стандартный ввод	стандартный вывод
2 0 2 10 20 10 7 10 13	2
1 1 2 13 13 15 6 5 16	1

Замечание

В первом примере имеются три слабых робота с ограничениями по весу 6, 2 и 9, а также два маленьких робота с ограничениями по размеру 4 и 7.

Все игрушки можно убрать за три минуты. Сделать это быстрее невозможно.

Во втором примере игрушку с весом 5 и размером 3 не может убрать ни один робот.

Слабый робот с наибольшим ограничением по весу имеет значение 5, однако вес игрушки должен быть строго меньше ограничения. Маленький робот имеет ограничение по размеру 2, однако размер игрушки равен 3. Поэтому убрать все игрушки невозможно.

Задача D. Справедливый лифт

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

Существует здание, состоящее из $2N$ этажей, пронумерованных числами от 1 до $2N$ снизу вверх.

Лифт в этом здании ровно один раз проехал с первого этажа до этажа $2N$.

Во время этой поездки лифтом воспользовались N человек. Человек i , где $1 \leq i \leq N$, вошёл в лифт на этаже A_i и вышел из него на этаже B_i .

При этом выполнялись следующие условия:

- $1 \leq A_i < B_i \leq 2N$;
- на каждом этаже ровно один человек либо входил в лифт, либо выходил из него.

Кроме того, пассажиры были довольно требовательными, поэтому выполнялось следующее условие.

Для человека i обозначим через C_i количество случаев, когда другой человек входил в лифт или выходил из него в то время, пока человек i находился в лифте. Иными словами,

$$C_i = B_i - A_i - 1.$$

Если существовал момент времени, когда люди i и j одновременно находились в лифте, то должно было выполняться равенство

$$C_i = C_j.$$

Значения последовательностей A и B были записаны, однако часть записей оказалась потеряна.

Если значение A_i или B_i было потеряно, во входных данных вместо него записано число -1 .

Кроме того, некоторые из сохранившихся записей также могут быть ошибочными.

Определите, существуют ли последовательности A и B , которые удовлетворяют всем описанным условиям и не противоречат сохранившимся записям.

Формат входных данных

В первой строке содержится одно целое число N — количество пассажиров.

В следующих N строках содержатся по два целых числа A_i и B_i — известные значения этажей, на которых человек i вошёл в лифт и вышел из него.

Значение -1 означает, что соответствующая запись была потеряна.

Гарантируется, что выполняются следующие ограничения:

- $1 \leq N \leq 100$;
- $A_i = -1$ или $1 \leq A_i \leq 2N$;
- $B_i = -1$ или $1 \leq B_i \leq 2N$;
- все числа во входных данных являются целыми.

Формат выходных данных

Выведите Yes, если существуют последовательности A и B , удовлетворяющие всем условиям и не противоречащие сохранившимся записям.

В противном случае выведите No.

Примеры

стандартный ввод	стандартный вывод
3 1 -1 -1 4 -1 6	Yes
2 1 4 2 3	No
2 4 1 2 4	No

Замечание

В первом примере можно положить $B_1 = 3$, $A_2 = 2$ и $A_3 = 5$.

Тогда все условия будут выполнены.

Люди 1 и 2 некоторое время одновременно находились в лифте, но это допустимо, поскольку $C_1 = C_2 = 1$.

Во втором примере люди 1 и 2 некоторое время одновременно находились в лифте. Однако $C_1 = 2$, а $C_2 = 0$, поэтому хотя бы одна из записей является ошибочной.

В третьем примере все записи сохранились, однако они явно ошибочны, поскольку человек 1 не мог войти в лифт на этаже 4 и выйти из него на этаже 1.

Задача Е. Дороги Катана

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

На острове Катан начинается большая строительная кампания. Несколько поселений решили объединить свои ресурсы, чтобы построить сеть новых дорог.

Для каждого из N поселений заранее составлен план действий, записанный строкой из символов (и). Символ (означает, что поселение привозит на общий склад один набор ресурсов, а символ) — что один набор ресурсов забирают со склада для строительства дороги.

Планы поселений можно выполнять в любом порядке, однако менять порядок действий внутри отдельного плана нельзя. Во время выполнения планов нельзя забирать ресурсы с пустого склада. Кроме того, после выполнения всех планов на складе не должно остаться ресурсов.

Определите, можно ли выбрать порядок выполнения планов поселений так, чтобы строительная кампания завершилась успешно.

Формат входных данных

В первой строке дано целое число N — количество поселений ($1 \leq N \leq 10^6$).

В следующих N строках даны строки $S_1, S_2, \dots, S_N$, описывающие планы поселений. Каждая строка непуста и состоит только из символов (и).

Суммарная длина всех строк не превосходит 10^6 .

Формат выходных данных

Выведите Yes, если планы поселений можно выполнить в некотором порядке так, чтобы склад ни в какой момент не оказался пустым при попытке забрать ресурсы, а после выполнения всех планов на складе не осталось ресурсов.

В противном случае выведите No.

Регистр букв в ответе не имеет значения.

Примеры

стандартный ввод	стандартный вывод
2) ((	Yes
2)((	No
4 ((((((()))) (())	Yes
3 ((())	No

Задача F. Результаты турнирной сетки

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

Рассмотрим строки, состоящие из символов $(,), [\text{ и }]$.

Строка x называется *правильной скобочной последовательностью*, если выполняется хотя бы одно из следующих условий:

- x является пустой строкой;
- существует правильная скобочная последовательность s , такая что x получается конкатенацией символа $($, строки s и символа $)$;
- существует правильная скобочная последовательность s , такая что x получается конкатенацией символа $[$, строки s и символа $]$;
- существуют непустые правильные скобочные последовательности s и t , такие что x получается конкатенацией строк s и t .

Например, строки $[]$, $([()])$ и $()[]$ являются правильными скобочными последовательностями, а строки $()$ и $[]$ правильными скобочными последовательностями не являются.

Дано чётное число N , а также две последовательности целых чисел A и B , каждая длины N .

Рассмотрим правильную скобочную последовательность $s = s_1 s_2 \dots s_N$ длины N . Определим её стоимость следующим образом:

- стоимость последовательности s равна сумме стоимостей всех её символов;
- стоимость символа на позиции i , где $1 \leq i \leq N$, равна A_i , если s_i равен $($ или $)$;
- стоимость символа на позиции i равна B_i , если s_i равен $[$ или $]$.

Найдите максимально возможную стоимость правильной скобочной последовательности длины N .

Формат входных данных

В первой строке содержится одно целое число N .

Во второй строке содержатся N целых чисел $A_1, A_2, \dots, A_N$.

В третьей строке содержатся N целых чисел $B_1, B_2, \dots, B_N$.

Гарантируется, что выполняются следующие ограничения:

- $2 \leq N \leq 10^5$;
- N является чётным;
- $1 \leq A_i \leq 10^9$ для всех $1 \leq i \leq N$;
- $1 \leq B_i \leq 10^9$ для всех $1 \leq i \leq N$.

Формат выходных данных

Выведите одно целое число — максимально возможную стоимость правильной скобочной последовательности длины N .

Примеры

стандартный ввод	стандартный вывод
4 4 2 3 1 2 3 2 4	12
10 866111664 844917655 383133839 353498483 472381277 550309930 378371075 304570952 955719384 705445072 178537096 218662351 231371336 865935868 579910117 62731178 681212831 16537461 267238505 318106937	6629738472

Замечание

В первом примере можно выбрать последовательность $() []$.

Её стоимость равна $A_1 + A_2 + B_3 + B_4 = 4 + 2 + 2 + 4 = 12$.

Можно доказать, что правильной скобочной последовательности с большей стоимостью не существует.

Задача G. Сорняк

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мегабайта

В саду Такахаси и Аоки растут N сорняков, пронумерованных от 1 до N . Высота сорняка i равна A_i .

Такахаси и Аоки решили вырвать все сорняки следующим способом.

Сначала Аоки выбирает не более K сорняков и вырывает их.

После этого Такахаси повторяет следующую операцию до тех пор, пока в саду не останется сорняков:

- Пусть H — максимальная высота среди оставшихся сорняков. Такахаси одновременно вырывает все сорняки, высота которых строго больше $\frac{H}{2}$.

Аоки хочет минимизировать количество операций, которые выполнит Такахаси.

Среди всех способов добиться минимального количества операций Такахаси Аоки также хочет вырвать как можно меньше сорняков.

Определите минимальное количество операций Такахаси и минимальное количество сорняков, которые должен вырвать Аоки для достижения этого количества операций.

Числа необходимо вывести в указанном порядке через пробел.

Формат входных данных

В первой строке содержатся два целых числа N и K — количество сорняков и максимальное количество сорняков, которое может вырвать Аоки.

Во второй строке содержатся N целых чисел $A_1, A_2, \dots, A_N$ — высоты сорняков.

Гарантируется, что выполняются следующие ограничения:

- $1 \leq N \leq 2 \cdot 10^5$;
- $0 \leq K \leq N$;
- $1 \leq A_i \leq 10^9$;
- все входные данные являются целыми числами.

Формат выходных данных

Выведите два целых числа:

- минимальное количество операций, которые выполнит Такахаси;
- минимальное количество сорняков, которые должен вырвать Аоки, чтобы достичь этого количества операций.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4 9	2 1
3 3 2 3 5	0 3
9 8 137 55 56 60 27 28 133 56 55	1 4

Замечание

В первом примере Аоки может выбрать сорняк 4 высотой 9 и вырвать его.

После этого максимальная высота среди оставшихся сорняков равна 4. Так как $\frac{4}{2} = 2$, за первую операцию Такахаси вырвет сорняки высоты 3 и 4.

Во время второй операции Такахаси вырвет оставшийся сорняк высоты 2. Таким образом, ему потребуется две операции.

Независимо от того, какой один сорняк выберет Аоки, Такахаси не сможет закончить работу за одну операцию.

Кроме того, если Аоки не вырвет ни одного сорняка, Такахаси потребуется три операции. Поэтому для достижения минимального количества операций Аоки должен вырвать как минимум один сорняк.

Во втором примере Аоки может вырвать все три сорняка. После этого Такахаси не потребуется выполнять ни одной операции.

Задача Н. Алхимический конвейер

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

В огромной алхимической лаборатории установлен конвейер из n магических устройств. Через устройства последовательно проходит кристалл, начальная энергия которого равна 1.

Каждое устройство имеет один из двух типов:

- $(+, a_i)$ — увеличивает энергию кристалла на a_i ;
- $(*, a_i)$ — умножает энергию кристалла на a_i .

Иными словами, если перед обработкой энергия кристалла была равна x , то после прохождения через устройство:

- типа $(+, a_i)$ она станет равна $x + a_i$;
- типа $(*, a_i)$ она станет равна $x \cdot a_i$.

Кристалл проходит через устройства в порядке их расположения на конвейере: результат работы первого устройства поступает во второе, результат работы второго — в третье и так далее.

Сейчас лаборатория переживает не лучшие времена, поэтому при текущем порядке устройств итоговая энергия кристалла не превосходит $2 \cdot 10^9$.

Совет алхимиков выделил вам b магических монет на перестройку конвейера.

За одну операцию можно выбрать любое устройство, извлечь его из текущего места и вставить в любое другое место конвейера. При этом взаимный порядок всех остальных устройств не изменяется.

Стоимость перемещения зависит от типа устройства:

- перемещение устройства вида $(+, a_i)$ стоит p монет;
- перемещение устройства вида $(*, a_i)$ стоит m монет.

Определите максимальную итоговую энергию кристалла, которую можно получить, потратив на перестройку конвейера не более b монет.

Формат входных данных

В первой строке записаны четыре целых числа n , b , p и m ($1 \leq n \leq 10^6$, $1 \leq b, p, m \leq 10^9$) — количество устройств, доступный бюджет, стоимость перемещения устройства типа «прибавление» и стоимость перемещения устройства типа «умножение» соответственно.

В следующих n строках записаны описания устройств в порядке их расположения на конвейере.

Описание каждого устройства начинается с символа $+$ или $*$, обозначающего его тип. После символа записано целое число a_i ($1 \leq a_i \leq 2 \cdot 10^9$).

Гарантируется, что при исходном порядке устройств итоговая энергия кристалла не превосходит $2 \cdot 10^9$.

Формат выходных данных

Выведите одно целое число — максимальную итоговую энергию кристалла, которую можно получить, потратив на перемещение устройств не более b монет.

Примеры

стандартный ввод	стандартный вывод
3 2 1 3 * 2 + 1 + 1	6
4 2 2 2 * 2 + 1 * 3 + 2	21
8 2 1 1 * 2 + 1 * 4 + 1 + 1 + 1 * 5 + 3	240

Замечание

В первом примере бюджета недостаточно, чтобы переместить устройство $(*, 2)$. Однако можно перенести оба устройства $(+, 1)$ в начало конвейера.

В результате получится следующий порядок устройств:

$(+, 1), (+, 1), (*, 2)$.

Энергия кристалла будет изменяться следующим образом:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 6$.

Во втором примере можно переместить только одно устройство. Перенесём устройство $(+, 2)$ из конца конвейера в его начало.

В результате получится следующий порядок устройств:

$(+, 2), (*, 2), (+, 1), (*, 3)$.

Энергия кристалла будет изменяться следующим образом:

$1 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 21$.

В третьем примере переместим устройство $(*, 4)$ так, чтобы оно находилось перед устройством $(*, 5)$, а устройство $(+, 3)$ перенесём в начало конвейера.

В результате получится следующий порядок устройств:

$(+, 3), (*, 2), (+, 1), (+, 1), (+, 1), (+, 1), (*, 4), (*, 5)$.

Энергия кристалла будет изменяться следующим образом:

$1 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 48 \rightarrow 240$.

Задача I. Парад волшебных фонарей

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

В канун Нового года в городе проходит парад волшебных фонарей. Всего в параде участвуют n фонарей, пронумерованных от 1 до n . Фонари зажигают по очереди в порядке их номеров.

Перед началом парада у организаторов есть не более k магических кристаллов. В i -й фонарь можно поместить b_i кристаллов, причём этот фонарь способен вместить не более a_i кристаллов. Таким образом, должны выполняться условия

$$0 \leq b_i \leq a_i$$

для всех i , а также

$$\sum_{i=1}^n b_i \leq k.$$

Яркость фонаря равна количеству помещённых в него кристаллов. После того как были зажжены первые j фонарей, площадь освещает самый яркий из них. Поэтому яркость площади в этот момент равна

$$\max(b_1, b_2, \dots, b_j).$$

Красота парада равна сумме яркостей площади после зажигания каждого фонаря, то есть

$$\sum_{j=1}^n \max(b_1, b_2, \dots, b_j).$$

Выберите целые значения $b_1, b_2, \dots, b_n$ так, чтобы красота парада была максимальной.

Обратите внимание, что в некоторые фонари можно не помещать ни одного кристалла.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и k ($1 \leq n \leq 10^5$, $1 \leq k \leq 360$) — количество фонарей и максимальное количество доступных кристаллов соответственно.

Вторая строка каждого набора входных данных содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq k$) — максимальное количество кристаллов, которое можно поместить в каждый фонарь.

Гарантируется, что сумма n по всем наборам входных данных не превышает $5 \cdot 10^5$, а сумма k по всем наборам входных данных не превышает 1800.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — максимальную красоту парада.

Пример

стандартный ввод	стандартный вывод
4	1
3 4	20
0 0 1	5
6 8	19
1 2 0 5 1 8	
3 4	
1 0 4	
5 8	
2 4 5 4 3	

Замечание

В первом примере единственный оптимальный способ распределить кристаллы по фонарям — выбрать $b = [0, 0, 1]$.

В третьем примере распределение $b = [1, 0, 4]$ недопустимо, поскольку для него требуется $1 + 0 + 4 = 5$ кристаллов, а организаторам доступно всего $k = 4$ кристалла.

В четвёртом примере один из оптимальных вариантов распределения кристаллов имеет вид $b = [2, 0, 5, 0, 0]$. После последовательного зажигания фонарей яркость площади будет равна соответственно 2, 2, 5, 5 и 5. Поэтому красота парада составит

$$2 + 2 + 5 + 5 + 5 = 19.$$

Задача J. Минимальное покрытие отрезков

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

Даны n отрезков в формате $[l; r]$ на числовой прямой.

Также даны m запросов в формате $[x; y]$. Какое минимальное число отрезков надо взять так, чтобы каждая точка (не обязательно целочисленная) от x до y была покрыта хотя бы одним из них?

Если нельзя выбрать отрезки так, чтобы каждая точка от x до y была покрыта, тогда выведите -1 на этот запрос.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — количество отрезков и количество запросов, соответственно.

В каждой из следующих n строк записаны по два целых числа l_i и r_i ($0 \leq l_i < r_i \leq 5 \cdot 10^5$) — данные отрезки.

В каждой из следующих m строк записаны по два целых числа x_i и y_i ($0 \leq x_i < y_i \leq 5 \cdot 10^5$) — запросы.

Формат выходных данных

Выведите m целых чисел. i -е число должно быть ответом на i -й запрос: либо минимальное число отрезков необходимое, чтобы каждая точка (не обязательно целочисленная) от x_i до y_i была покрыта хотя бы одним из них, либо -1 , если нельзя выбрать отрезки так, чтобы каждая точка от x_i до y_i была покрыта.

Примеры

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

Замечание

В первом примере три запроса:

- запрос $[1; 3]$ может быть покрыт отрезком $[1; 3]$;
- запрос $[1; 4]$ может быть покрыт отрезками $[1; 3]$ и $[2; 4]$. Нельзя покрыть $[1; 4]$ одним отрезком;
- запрос $[3; 4]$ может быть покрыт отрезком $[2; 4]$; Не важно, что покрыты какие-то точки вне данного запроса.

Во втором примере четыре запроса:

1. запрос $[1; 2]$ может быть покрыт отрезком $[1; 3]$. Обратите внимание, что можно выбрать любой из отрезков $[1; 3]$;
2. запрос $[1; 3]$ может быть покрыт отрезком $[1; 3]$;
3. запрос $[1; 4]$ не может быть покрыт никаким набором отрезков;
4. запрос $[1; 5]$ не может быть покрыт никаким набором отрезков. Обратите внимание, что отрезки $[1; 3]$ и $[4; 5]$ вместе не покрывают $[1; 5]$, потому что даже нецелые точки должны быть покрыты. Здесь 3.5 не покрыта, например.