

## Задача А. Найти вершину

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

Вам дано дерево — связный неориентированный граф без циклов. В нём загадали одну из вершин. Вы можете задавать вопросы интерактору: за один вопрос вы можете выбрать ребро и узнать, какой из двух концов ребра находится ближе к загаданной вершине, то есть, до какого из двух концов меньше кратчайшее расстояние от загаданной вершины. Вам требуется определить загаданную вершину за минимальное для данного дерева число запросов в худшем случае.

Обратите внимание, что загаданная вершина может быть не фиксирована интерактором заранее: в зависимости от ваших запросов он может менять её на любую другую, с условием, что это не противоречит ответам на предыдущие запросы.

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $2 \leq n \leq 100$ ) — количество вершин в дереве.

В каждой из следующих  $n-1$  строк записаны два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ), обозначающих ребро между вершинами  $u$  и  $v$ . Гарантируется, что данные ребра образуют дерево.

### Протокол взаимодействия

После считывания выходных данных вы можете отправлять интерактору запросы двух типов:

- «?  $u$   $v$ » — узнать для ребра дерева  $(u, v)$  ( $1 \leq u, v \leq n$ ), какой из его концов ближе к загаданной вершине. В ответ вы получите одно число: номер одной из двух вершин. Обратите внимание, что  $u$  и  $v$  должны быть соединены ребром в дереве, поэтому они не могут находиться на равном расстоянии от загаданной вершины.
- «!  $u$ » — сообщить интерактору о том, что вы выяснили номер загаданной вершины. После вывода этой команды ваша программа должна немедленно завершиться.

После каждого вопроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт `Idleness Limit Exceeded` или **Превышено реальное время работы**. Для сброса буфера вы можете использовать:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `sys.stdout.flush()` в Python;
- смотрите документацию для других языков.

В случае, если вы сделаете большее число запросов, чем может быть необходимо для данного дерева в худшем случае, вы получите вердикт **Неправильный ответ**.

## Примеры

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

## Задача В. Фэйк Ньюз

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

В Бйитландии зарегистрировано несколько информационных агенств, в том числе новое информационное агенство ООО «неверлай дот ком». Начальник этой организации хочет выпустить сенсационное расследование. Последний месяц он ждет подходящего момента, чтобы новость стала как можно более популярной.

У каждого агенства есть свой показатель авторитетности  $A$ , который может отличаться для разных агенств. Процесс публикации новости в Бйитландии устроен так: первым делом агенство ООО «неверлай дот ком» публикует новость. Это запускает цепную реакцию, устроенную следующим образом:

- Каждое агенство из тех, кто еще не опубликовал данную новость, вычисляет число  $P$ : количество других агенств, опубликовавших новость.
- Если агенство имеет показатель авторитетности  $A$ , а число уже опубликовавших новость агенств  $P$  не меньше  $A$ , то это агенство публикует данную новость.
- Процесс повторяется до тех пор, пока предыдущее условие выполняется хотя бы для одного из агенств.

ООО «неверлай дот ком» хочет выбрать момент, когда им стоит опубликовать новость, но в Бйитландии часто появляются новые информационные агенства, а старые часто закрываются. От вас требуется обрабатывать запросы двух типов:

1. «+  $A$ » появляется новое агенство с показателем авторитетности  $A$
2. «-  $A$ » одно из агенств с показателем авторитетности  $A$  прекращает работу.

Для запросов второго типа гарантируется, что существовало хотя бы одно СМИ с показателем авторитетности  $A$ . Обратите внимание, что в один момент может существовать несколько агенств с одинаковым показателем авторитетности  $A$ .

После каждого запроса вам требуется определить, сколько агенств опубликуют новость, если ООО «неверлай дот ком» опубликует ее прямо сейчас.

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 500\,000$ ) — количество запросов.

В следующих  $n$  строках вводятся запросы в формате, описанном в условии задачи: сначала задаётся один из символов «+», «-», затем следует целое число  $A$  — показатель авторитетности ( $1 \leq A \leq 500\,000$ ).

### Формат выходных данных

Выведите  $n$  строк. В каждой строке должно быть одно число — количество агенств, которые опубликуют новость от ООО «неверлай дот ком» после  $i$ -го запроса.

## Примеры

стандартный ввод	стандартный вывод
5 + 4 + 1 + 3 + 2 - 3	1 2 2 5 3
8 + 1 + 1 + 2 - 1 + 4 - 1 + 1 + 1	2 3 4 3 3 1 3 5

## Замечание

Рассмотрим первый пример:

- После первого запроса ни одно другое агенство не будет публиковать новость от ООО «неверлай дот ком».
- После второго запроса появилось агенство, которое готово опубликовать новость.
- После третьего запроса все еще лишь одно агенство готово опубликовать новость.
- После четвёртого запроса процесс публикации будет устроен следующим образом:
  1. ООО «неверлай дот ком» публикует новость.
  2. агенство с  $A = 1$  публикует новость.
  3. агенство с  $A = 2$  публикует новость.
  4. агенство с  $A = 3$  публикует новость.
  5. агенство с  $A = 4$  публикует новость.
- После последнего запроса процесс публикации остановится на агенстве с  $A = 2$ .

## Задача С. Отборочный этап

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В одной очень известной олимпиаде участвуют более ста человек. Олимпиада состоит из двух этапов: отборочного и заключительного. В заключительный этап проходят хотя бы сто участников. Отборочный же этап состоит из двух туров.

В каждом туре участников упорядочивают по невозрастанию баллов. В случае, если два участника набрали одинаковый балл, они упорядочиваются по номеру паспорта. В соответствии с законодательством номера паспортов всех участников различны.

В первом туре участник на сотом месте набрал  $a$  баллов, при этом все участники первого тура, занявшие места от первого до сотого включительно, набрали не менее  $b$  баллов во втором туре.

Во втором туре участник на сотом месте набрал  $c$  баллов, при этом все участники второго тура, занявшие места от первого до сотого включительно, набрали не менее  $d$  баллов в первом туре.

Упорядочим всех участников по невозрастанию суммы их баллов за два тура. При равенстве результатов упорядочим участников по номеру паспорта. Тогда проходной балл для попадания в заключительный этап — это сумма баллов за оба тура у участника на сотом месте.

Помогите жюри узнать, каким может быть минимальный проходной балл на заключительный этап.

### Формат входных данных

В единственной строке содержатся четыре целых числа  $a$ ,  $b$ ,  $c$  и  $d$  ( $0 \leq a, b, c, d \leq 100$ ;  $d \leq a$ ;  $b \leq c$ ). Можно показать, что для любых входных данных, удовлетворяющих ограничениям из условия, существует хотя бы один корректный вариант проведения олимпиады.

### Формат выходных данных

Выведите одно целое число — ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
1 2 2 1	3
4 8 9 2	12

### Замечание

В первом примере на олимпиаду могут отбираться сто один человек с баллами 1 и 2 за первый и второй тур, соответственно. Сумма баллов у сотого участника будет равна 3.

Во втором примере на олимпиаду могло отбираться пятьдесят человек с баллами 5 и 10, пятьдесят человек с баллами 4 и 8 и пятьдесят человек с баллами 2 и 9, соответственно. Сумма баллов у сотого участника будет равна 12.

## Задача D. Сборы в поход

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	768 мегабайт

Вы собираетесь пойти в поход. Для этого вам нужно разложить всё снаряжение по нескольким рюкзакам. У вас есть неограниченное количество рюкзаков вместительностью  $s$  литров каждый.

Всем известно, что когда идёшь в поход, вещи каждого типа нужно брать в нескольких экземплярах, чтобы если одна промокнет, была запасная. Разумеется, разные вещи одного типа нужно класть в разные рюкзаки.

Вы решили взять в поход вещи  $n$  типов, причём вещей типа  $i$  вы решили взять  $k_i$  штук. Каждая вещь имеет свой объем —  $j$ -я вещь типа  $i$  имеет объём  $w_{i,j}$ . Вы хотите, используя минимальное число рюкзаков, разложить все имеющиеся вещи по ним так, чтобы суммарный объем вещей в каждом рюкзаке не превышал  $s$ , а также чтобы ни в одном из рюкзаков не оказалось двух вещей одного типа. Определите минимальное количество рюкзаков, которое вам понадобится, чтобы пойти в поход.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $s$  ( $1 \leq n \leq 23$ ,  $1 \leq s \leq 10^{16}$ ) — количество типов предметов и вместительность каждого из рюкзаков.

В следующих  $n$  строках даны описания вещей. В  $i$ -й строке содержится описание вещей  $i$ -го типа. Описание начинается с одного целого числа  $k_i$  ( $1 \leq k_i \leq 23$ ) — количества вещей типа  $i$ . Затем следуют  $k_i$  целых чисел  $w_{i,j}$  ( $1 \leq w_{i,j} \leq s$ ) — объёмы вещей  $i$ -го типа.

Суммарное количество вещей не превышает 23.

### Формат выходных данных

Выведите минимальное количество рюкзаков, которое необходимо, чтобы разместить все вещи по описанным выше правилам.

### Пример

стандартный ввод	стандартный вывод
3 70 3 10 30 40 2 20 60 1 50	4

### Замечание

Рассмотрим пример из условия. В нём есть вещи трёх типов: наверняка это были укулеле, бутылки с водой и гири.

В первый рюкзак можно положить укулеле объёмом 40 литров и бутылку с водой объёмом 20 литров. Во второй рюкзак можно положить укулеле объёмом 10 литров и бутылку с водой объёмом 60 литров. В третий рюкзак можно положить укулеле объёмом 30 литров. В четвёртом рюкзаке будет лежать гиря объёмом 50 литров. Можно показать, что разложить нужные вещи по трём рюкзакам не получится.

## Задача Е. Деление

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Из предметов в школе Олегу больше всего нравятся история и математика, а его любимый раздел математики — деление. Чтобы улучшить свои навыки в делении, Олег загадал два целых числа  $p$  и  $q$  и решил найти максимальное число  $x$ , такое что  $p$  делится нацело на  $x$ , а  $x$  не делится нацело на  $q$ . Так как Олег очень хорош в делении, то он быстро нашёл нужное число. Теперь ему интересно, сможете ли вы тоже справиться с этой задачей.

### Формат входных данных

В первой строке задано целое число  $p$  ( $1 \leq p \leq 10^{100}$ ).

Во второй строке задано целое число  $q$  ( $2 \leq q \leq 10^{12}$ ).

Можно показать, что при данных ограничениях существует хотя бы одно подходящее  $x$ .

### Формат выходных данных

Выведите максимальное число  $x$ , такое что  $p$  делится нацело на  $x$ , а  $x$  не делится на  $q$ .

### Примеры

стандартный ввод	стандартный вывод
10 4	10
12 6	4
179 822	179

### Замечание

В первом примере число  $x = 10$  подходит, так как это максимальный делитель 10, и 10 не делится на 4.

Во втором примере заметим, что число 12 не подходит в качестве  $x$ , так как 12 делится на 6, 6 не подходит в качестве  $x$ , так как 6 делится на 6, а следующий по величине делитель 12 — это 4, и  $x = 4$  нам подходит, так как 4 не делится нацело на 6.

## Задача F. Парад во время чумы

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

В далекой стране Байтландии сегодня проходит парад. Военные машины уже проехали по городу, самолёты уже пролетели, остался лишь марш солдат, который вам и предстоит организовать в кратчайшие сроки.

Всего в марше должны были участвовать  $n$  отрядов солдат, по  $m$  солдат в каждом. К сожалению, собираться большим числом людей в наши времена крайне сложно, и вы решили, что хотите выбрать из  $n$  отрядов по одному представителю и пройти одной большой шеренгой длины  $n$ . Выбранные солдаты будут расположены в шеренге в порядке возрастания номеров их отрядов.

Ваше начальство хочет, чтобы строй был максимально ровным из возможных. *Неровность* шеренги определяется максимальным модулем разности роста соседних солдат.

Время поджимает — выходить уже через 5 часов! Определите, какой минимальной неровности шеренги можно добиться при таких условиях.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $m$  ( $1 \leq nm \leq 10^6$ ) — количество отрядов и число солдат в каждом из них.

В следующих  $n$  строках находится по  $m$  целых чисел  $a_{i,j}$  ( $10^6 \leq a_{i,j} \leq 2 \cdot 10^6$ ) — высота в микрометрах  $j$ -го солдата  $i$ -го отряда.

### Формат выходных данных

Выведите одно число — минимальную возможную неровность шеренги в микрометрах.

### Пример

стандартный ввод	стандартный вывод
3 4 1830000 1790000 1810000 1820000 1730000 1690000 1750000 1760000 1910000 1850000 1800000 1710000	40000

### Замечание

В тесте из условия одним из оптимальных вариантов будет шеренга из солдат с ростами: 179 см, 176 см, 180 см. Модуль разности роста первого и второго солдат составляет 3 см, а второго и третьего — 4. Таким образом, неровность шеренги равна 4 см или 40000 микрометров.



## Задача G. Прямоугольная ломаная

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

На координатной плоскости была нарисована замкнутая ломаная, состоящая из вертикальных и горизонтальных отрезков (параллельных осям координат). Горизонтальные и вертикальные отрезки этой ломаной чередовались (за горизонтальным отрезком шёл вертикальный, и наоборот). В этой ломаной не было пересечений отрезков по внутренним точкам, то есть если какие-то два отрезка пересекались, то точка их пересечения являлась одним из концов каждого из них (обратите внимание на примеры в пояснении к условию).

К сожалению, ломаную стёрли, а про неё осталось совсем немного информации: известны лишь длины всех вертикальных и всех горизонтальных отрезков. Вам требуется восстановить любую ломаную, удовлетворяющую условию, или определить, что такой ломаной не существует.

### Формат входных данных

Вам нужно получить ответ для  $t$  наборов входных данных.

В первой строке задано целое число  $t$  ( $1 \leq t \leq 200$ ) — количество наборов входных данных. Затем следуют  $t$  наборов входных данных.

В первой строке описания набора входных данных содержится целое число  $h$  ( $1 \leq h \leq 1000$ ) — количество горизонтальных отрезков.

Во второй строке содержатся  $h$  целых чисел ( $1 \leq l_i \leq 1000$ ) — длины горизонтальных отрезков ломаной, в произвольном порядке.

В третьей строке содержится целое число  $v$  ( $1 \leq v \leq 1000$ ) — количество вертикальных отрезков.

В четвёртой строке содержатся  $v$  целых чисел ( $1 \leq p_i \leq 1000$ ) — длины вертикальных отрезков ломаной, в произвольном порядке.

Все тестовые наборы входных данных разделяются пустой строкой.

Сумма  $h + v$  по всем наборам тестовых данных не превосходит 1000.

### Формат выходных данных

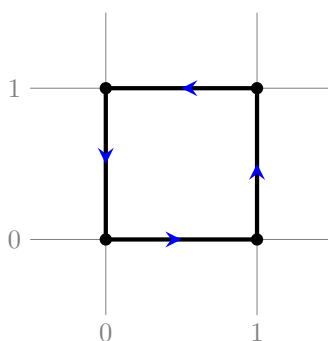
Для каждого набора входных данных выведите слово «Yes» (без кавычек), если требуемая ломаная существует, «No» в противном случае. Если ломаная существует, в следующих  $n$  строках выведите последовательно вершины этой ломаной. В  $i$ -й строке выведите два целых числа  $x_i, y_i$  — координаты  $i$ -й вершины. Обратите внимание, что все отрезки этой ломаной должны быть параллельны осям координат, при этом после горизонтального отрезка должен идти вертикальный, и наоборот. Все координаты не должны превосходить  $10^9$  по абсолютной величине.

## Примеры

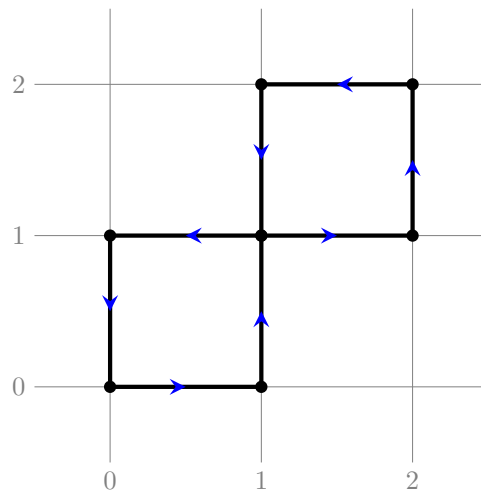
стандартный ввод	стандартный вывод
2 2 1 1 2 1 1  2 1 2 2 3 3	Yes 1 0 1 1 0 1 0 0 No  1 2 2 3 3
2 4 1 1 1 1 4 1 1 1 1  3 2 1 1 3 2 1 1	Yes 1 0 1 1 2 1 2 2 1 2 1 1 0 1 0 0 Yes 0 -2 2 -2 2 -1 1 -1 1 0 0 0
2 4 1 4 1 2 4 3 4 5 12  4 1 2 3 6 2 1 3	Yes 2 0 2 3 3 3 3 7 4 7 4 12 0 12 0 0 No

## Замечание

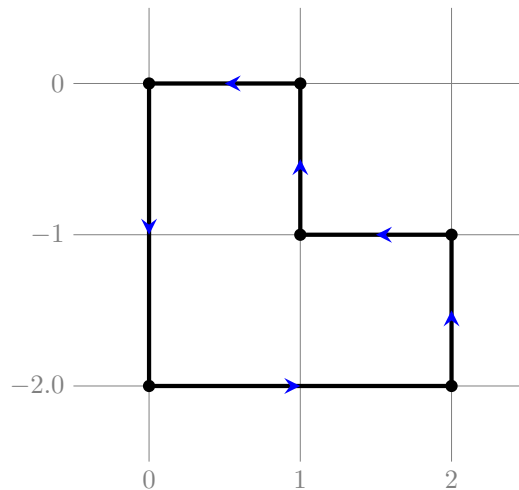
В первом примере первого теста ответ «Yes» — в качестве примера можно привести квадрат:



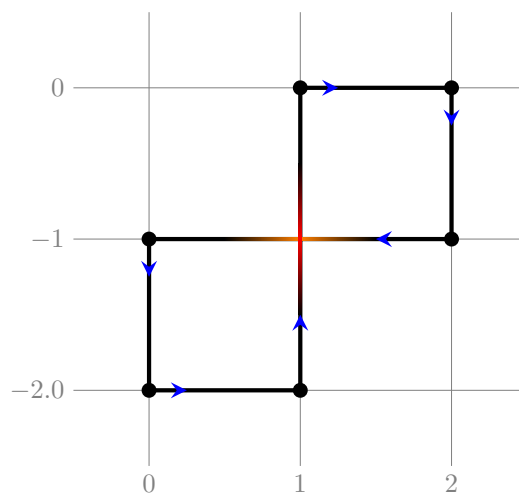
В первом примере второго теста искомая ломаная существует. Обратите внимание, что ломаная пересекается по вершинам-концам отрезков:



Во втором примере второго теста искомая ломаная может выглядеть, как на рисунке ниже:



Обратите внимание, что пример ниже не будет корректным для этого набора входных данных, так как содержит самопересечения:



## Задача Н. Тимбилдинг

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Начался новый учебный год, и в университет Берляндии пришли  $n$  новых студентов, разбитых на  $k$  групп, некоторые из которых могут быть пустыми. Среди студентов есть  $m$  пар знакомых, причём знакомые студенты могут быть как из одной, так и из разных групп. Алиса, как куратор нового набора, позвала всех новоприбывших на организационную встречу. На ней она хочет устроить игру, чтобы еще незнакомые студенты получше узнали друг друга. Для этого она выберет две группы, студенты из которых будут играть. При этом правила игры требуют разделить участников на две команды так, чтобы внутри каждой из команд никто не знал друг друга.

Алису интересует: сколько существует способов выбрать две различные группы студентов так, чтобы получилось сыграть в игру по всем правилам. При этом в игре должны участвовать все студенты обеих групп.

Обратите внимание, что команды, на которые Алиса разделит студентов, не обязаны совпадать с группами, в которых учатся участники.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n \leq 500\,000$ ,  $0 \leq m \leq 500\,000$ ,  $2 \leq k \leq 500\,000$ ) — количество студентов, количество пар знакомых студентов и количество групп, соответственно.

Во второй строке заданы  $n$  целых чисел  $c_i$  ( $1 \leq c_i \leq k$ ), где  $c_i$  равняется номеру группы, в которой учится  $i$ -й студент.

Далее следуют  $m$  строк. В  $i$ -й строке записаны два числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ), означающие, что  $a_i$ -й и  $b_i$ -й студент знают друг друга. Гарантируется, что  $a_i \neq b_i$ , и что если пара студентов знает друг друга, то она встречается ровно один раз.

### Формат выходных данных

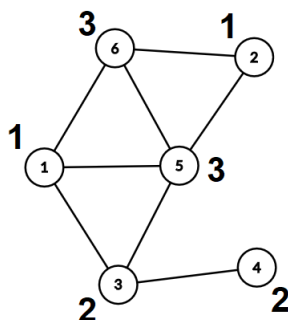
Выведите одно число — количество способов выбрать две различные группы так, чтобы получилось сыграть в игру по всем правилам.

## Примеры

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

## Замечание

Граф знакомств для первого тестового примера выглядит следующим образом (рядом с каждым студентом подписан номер его группы):



В данном случае нам подходят способы:

- Выбрать первую и вторую группу — например, можно отнести студентов номер 1 и 4 к первой команде, а студентов номер 2 и 3 ко второй.
- Выбрать вторую и третью группу — можно отнести студентов номер 3 и 6 к первой команде, а студентов номер 4 и 5 ко второй.

- Выбрать первую и третью группы мы не можем, потому что не существует разбиения на команды, удовлетворяющего правилам игры.

Во втором тестовом примере мы можем выбрать любую пару групп. Обратите внимание, что несмотря на то, что в третьей группе нет студентов, мы все равно можем ее выбирать.

## Задача I. ТВ

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

После тяжёлого рабочего дня Алексей решил расслабиться за просмотром телевизора. Всего есть  $n$  телеканалов, на  $i$ -м телеканале показывают интересную для Алексея передачу в отрезок времени  $[l_i, r_i]$ .

Так как жизнь Алексея расписана по минутам, то он смотрит телевизор по определённому алгоритму. Пусть в текущий момент Алексей смотрит телеканал  $x$  в момент времени  $t$ . Тогда:

- если на этом телеканале в момент времени  $t$  идёт интересная передача, то Алексей досматривает ее до конца и переходит к следующему телеканалу под номером  $x + 1$ ;
- если же в момент времени  $t$  на телеканале  $x$  нет интересной передачи, то Алексей моментально переходит к следующему телеканалу  $x + 1$ ;
- если же канала  $x + 1$  не существует (при  $x = n$ ), то Алексей заканчивает просмотр телевизора.

Алексей пока не определился, в какой момент и с какого телеканала начать просмотр. У него есть  $m$  вариантов,  $i$ -й вариант предполагает начало просмотра телевизора с телеканала  $x_i$  в момент времени  $t_i$ . Для каждого из вариантов определите время, которое Алексей проведет за просмотром интересных передач.

### Формат входных данных

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 300\,000$ ) — количество телеканалов.

Следующие  $n$  строк содержат описания телеканалов:  $i$ -я строка содержит два числа  $l_i$  и  $r_i$  ( $1 \leq l_i < r_i \leq 10^9$ ) — отрезок времени, на протяжении которого на  $i$ -м телеканале идёт интересная передача.

В следующей строке задано целое число  $m$  ( $1 \leq m \leq 300\,000$ ) — количество вариантов просмотра.

Следующие  $m$  строк содержат описания вариантов просмотра, в  $i$ -й строке содержатся два целых числа  $x_i$  и  $t_i$  ( $1 \leq x_i \leq n; 1 \leq t_i \leq 10^9$ ) — номер телеканала и момент времени, соответственно.

### Формат выходных данных

Для  $i$ -го варианта просмотра выведите одно целое число — суммарное время, которое Алексей будет смотреть интересные передачи, если начнет просмотр с телеканала  $x_i$  в момент времени  $t_i$ .

### Пример

стандартный ввод	стандартный вывод
3	2
1 3	0
2 4	3
7 10	0
4	
1 2	
2 5	
1 7	
3 10	

### Замечание

Рассмотрим тест из условия.

Если Алексей начнёт просмотр с первого телеканала в момент времени 2, то он будет смотреть на нём интересную передачу в течение одной минуты до момента времени 3, затем переключится на второй канал, на котором будет смотреть интересную передачу в течение одной минуты до момента

времени 4. В момент времени 4 Алексей переключится на третий канал, по которому в этот момент не идёт интересная передача, поэтому в этот момент он закончит просмотр телевизора.

Во втором варианте просмотра, в момент времени 5 интересная передача не идёт ни по одному из каналов, поэтому Алексей сразу завершит просмотр.

В третьем варианте просмотра Алексей начнёт просмотр с первого канала в момент времени 7, так что он переключится сразу на второй, а потом и сразу на третий канал, после чего будет смотреть на нём интересную передачу в течение трёх минут до момента времени 10, после чего завершит просмотр телевизора.



## Задача J. Разбивай и суммируй

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Вам дан массив  $a$  длины  $2n$ . Рассмотрим разбиение массива  $a$  на две подпоследовательности  $p$  и  $q$  длины  $n$ . Отсортируем  $p$  по неубыванию, а  $q$  — по невозрастанию, и получим последовательности  $x$  и  $y$ , соответственно. Тогда назовём *стоимостью* разбиения  $f(p, q) = \sum_{i=1}^n |x_i - y_i|$ .

Вам необходимо найти сумму  $f(p, q)$  по всем корректным разбиениям массива  $a$ . Так как это число может быть воистину огромным, требуется посчитать остаток от деления его на 998244353.

### Формат входных данных

В первой строке задано единственное целое число  $n$  ( $1 \leq n \leq 150\,000$ ).

Во второй строке заданы  $2n$  целых чисел  $a_1, a_2, \dots, a_{2n}$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива  $a$ .

### Формат выходных данных

Выведите одно целое число — ответ на задачу по модулю 998244353.

### Примеры

стандартный ввод	стандартный вывод
1 1 4	6
2 2 1 2 1	12
3 2 2 2 2 2 2	0

### Замечание

Два разбиения массива считаются различными, если отличаются наборы индексов элементов, входящих в подпоследовательность  $p$ .

В первом примере существует два корректных разбиения массива  $a$ :

- $p = [1], q = [4]$ , тогда  $x = [1], y = [4], f(p, q) = |1 - 4| = 3$
- $p = [4], q = [1]$ , тогда  $x = [4], y = [1], f(p, q) = |4 - 1| = 3$ .

Во втором примере существует шесть корректных разбиений массива  $a$ :

- $p = [2, 1], q = [2, 1]$  (в подпоследовательность  $p$  выбраны элементы с индексами 1 и 2 в исходном массиве).
- $p = [2, 2], q = [1, 1]$ .
- $p = [2, 1], q = [2, 1]$  (в подпоследовательность  $p$  выбраны элементы с индексами 1 и 4 в исходном массиве).
- $p = [1, 2], q = [2, 1]$ .
- $p = [1, 1], q = [2, 2]$ .
- $p = [2, 1], q = [2, 1]$  (в подпоследовательность  $p$  выбраны элементы с индексами 3 и 4 в исходном массиве).