

"А" Шоколадка

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

У Игната праздник! И по этому поводу ему подарили шоколадку размером $n \times n$ долек. В разгаре веселья он положил эту шоколадку на стол и на время забыл о ней. Вернувшись за ней, он понял, что случилось ужасное! Его шоколадку кто-то сломал! И, мало того, что сломал, так ещё и мог съесть несколько долек. Помогите Игнату выяснить всю правду: ел ли кто-то его шоколадку?

Сломанная шоколадка представляет собой две половинки — левую и правую. Левая половинка выровнена по левому краю, правая — по правому. Каждая половинка состоит из n рядов. В i -м ряду левой половинки l_i долек, правой — r_i долек. Игнат считает, что шоколадку никто не ел, если из левой и правой половинок можно сложить целую шоколадку $n \times n$, не поднимая их со стола и не переворачивая.

На картинке приведены шоколадки из первых двух запросов в примере.

На вход подается несколько запросов. В первой строке задано число t ($1 \leq t \leq 100$) — количество запросов, на которые необходимо ответить.

Формат входных данных

Далее, каждый в своей строке, заданы сами запросы в формате: сначала n ($2 \leq n \leq 100$) — размер шоколадки, далее n пар чисел: $l_1 r_1, l_2 r_2, \dots, l_n r_n$ ($1 \leq l_i, r_i < n, l_i + r_i \leq n$), все числа целые.

Формат выходных данных

Для каждого запроса в отдельной строке выведите «yes», если шоколадку кто-то ел, или — «no» в противном случае.

Входные данные Выходные данные

Примеры	3	
	3 2 1 1 2 2 1	no
	3 1 1 1 1 2 1	yes
	3 1 1 1 1 1 1	yes

"В" Игра

Ограничение по времени 2 секунды

Ограничение по памяти 256 мегабайт

Жюри Russian Code Cup иногда в свободное время играет в различные игры.

Например, между последними двумя раундами жюри придумало новую игру со следующими правилами.

- Изначально на бесконечном поле в точке (x_s, y_s) стоит фигура.
- За один ход игрок может передвинуть фигуру из клетки (x_1, y_1) в любую клетку

- (x_2, y_2) такую, что $|x_1 - x_2| + |y_1 - y_2| \leq m$.
- Игроют в игру два игрока, ходы совершаются по очереди.
- Выигрывает тот, кто сможет поставить фигуру в точку (xf, yf) .

Вам дано n описаний игр, каждое из которых представляет собой пять чисел: ограничение на дальность хода фигуры m , начальное и конечное положение фигуры. Необходимо для каждой игры выяснить, кто выиграет при оптимальной игре и на каком ходу. Ни один из игроков не хочет проигрывать, поэтому игроки могут играть бесконечно долго.

Формат входных данных	Первая строка содержит одно целое число n ($1 \leq n \leq 100$) — количество игр. Далее, в n строках задано по пять целых чисел: m, xs, ys, xf, yf (все числа лежат в диапазоне от 1 до 1000) — величина хода фигуры, начальные и конечные координаты фигуры, соответственно. Точки, задаваемые начальными и конечными координатами, различны.	
Формат выходных данных	Для каждого описания игры выведите в отдельной строке «First», если выигрывает первый, «Second», если выигрывает второй, и «Infinity», если игроки будут играть бесконечно долго. Если партия конечна, то выведите также на той же строке через пробел номер хода, после которого заканчивается игра.	
	Входные данные Выходные данные	
Примеры	2 2 3 1 3 2 2 3 1 1 4	First 1 Infinity

"С" Часы

Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Один из клиентов — кажется частный детектив — принес в часовую мастерскую часы в ужасном состоянии. Мало того, что они стоят, так еще кто-то залил их краской так, что видно не весь циферблат. Клиент попросил вас выяснить, во сколько часы остановились. Так как циферблат видно не полностью, вы с клиентом сошлись на том, что определите количество вариантов положения стрелок, которое могло быть на часах, когда произошла остановка.

Циферблат представляет собой круг, на который нанесены минутные деления от 0 до 59 с шагом один. У часов есть только часовая и минутная стрелка. В каждый момент каждая стрелка находится точно на некотором делении. Каждому часу соответствуют 5 делений. С нулевой по 11 минуту включительно часовая стрелка находится на первом делении часа, с 12 по 23 — на втором, с 24 по 35 — на третьем, с 36 по 47 — на четвертом, с 48 по 59 — на пятом. Например, если сейчас 3:30, то минутная стрелка находится на 30 делении, а часовая — на 17.

В переданных вам часам из-за пролитой на них краски осталась видна только часть циферблата внутри некоторого сектора. Определите количество различных вариантов положения стрелок, которое могло быть на часах когда они остановились.

На картинке показаны запросы из примера.

	В первой строке находится количество запросов q ($1 \leq q \leq 1000$). Далее следуют запросы по одному в строке.	
Формат входных данных	Каждый запрос задается четырьмя целыми числами a, b, h и m , записанными через пробел ($0 \leq a, b \leq 59$) — отрезок видимых значений. Это означает, что видна только та часть циферблата, по которой проходит минутная стрелка от a -й до b -й минуты, включительно. В частности, если $a \leq b$, то виден отрезок $[a; b]$, а если $b < a$, то виден отрезок $[a; 59] \cup [0; b]$.	
Формат выходных данных	h отвечает за положение часовой стрелки. Если $h = -1$, то часовую стрелку не видно. Иначе, h лежит в рассмотренном выше отрезке от a до b . Значение m аналогичным образом отвечает за положение минутной стрелки.	
Примеры	Входные данные 2 55 5 0 0 32 32 -1 32	Выходные данные 1 11

"D" ГАС «Очередь»

Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

За последние несколько лет электронные очереди прочно вошли в повседневную жизнь. Во многих государственных учреждениях можно встретить терминал, печатающий бумажку с номером, и, не задавая привычный вопрос «Кто последний?», посетители с помощью электронного табло узнают, сколько еще им ждать и когда наступит их очередь.

Однако, пока такие системы далеки от совершенства. Например, вызывает вопросы стандартный принцип любой очереди: «Первым обслуживается тот, кто первым пришел». При разработке инновационной ГАС «Очередь» решено было сделать ее такой, что выполнение этого принципа не требуется. Вместо этого, новая система призвана минимизировать количество негатива, приходящегося на чиновника, на прием к которому стоят люди в очереди.

Известно, что у каждого человека есть такой критерий, как раздражительность. Если этот параметр равен w , то через t часов ожидания в очереди этот человек обрушит на голову чиновника ровно wt единиц злобы и ругани. Так, если посетителя начнут обслуживать сразу же после его прихода, то чиновник не пострадает, а если посетитель пришел в начале третьего часа, а его обслуживание началось только в начале пятого — количество

гнева будет равно $2w$.

Также известно, что на обслуживание каждого посетителя уходит ровно час, а каждый посетитель приходит в начале какого-либо часа. Ваша задача заключается в том, чтобы по данным вам показателям раздражительности и временам прихода посетителей определить, сколько негатива достанется чиновнику при оптимальном порядке обслуживания клиентов.

В первой строке задано одно целое число t — количество случаев, которые вам предстоит обработать. Далее следуют t описаний самих случаев.

Формат
входных
данных

Описание каждого случая состоит из: числа n в первой строке — количества посетителей, и n описаний самих посетителей. Для каждого посетителя в отдельной строке указаны два целых числа r_i и w_i ($1 \leq r_i, w_i \leq 10^6$) — номер часа, в начале которого посетитель пришел, и его коэффициент раздражительности, соответственно.

Суммарное количество посетителей во всех случаях одного теста не превосходит 10^5 .

Формат
выходных
данных

Для каждого случая в отдельной строке выведите ответ — минимальное суммарное количество негатива, которое получит чиновник.

Входные данные Выходные данные

Примеры

```
2
3
1 3
1 3
1 3
3
1 3
2 5
1 4
```

9
6

"Е" Помехи

Ограничение
по времени

2 секунды

Ограничение
по памяти

256 мегабайт

При передаче информационных сообщений по каналам связи часто возникают ошибки, и получается что полученное сообщение отличается от отправленного. Для борьбы с этим применяют различные коды обнаружения ошибок, а также корректирующие коды, позволяющие исправлять наиболее вероятные ошибки. Одним из методов обнаружения ошибок является передача вместе с сообщением некоторого хеша — контрольной суммы, которую можно рассчитать для полученного сообщения и сравнить с контрольной суммой исходного сообщения.

В этой задаче вам необходимо будет реализовать коррекцию ошибок в полученном сообщении с помощью контрольной суммы, которая рассчитывается как полиномиальный хеш. Будем считать, что передается некоторая битовая строка, хранящаяся как двоичное

число из n бит. Если пронумеровать биты числа с нуля, начиная с младшего, то полиномиальный хеш рассчитывается по следующей формуле:

$$h(a) = (a_0 + a_1 t + a_2 t^2 + \dots + a_i t^i + \dots + a_{n-1} t^{n-1}) \bmod M, \text{ где } a_i \text{ — } i\text{-й бит числа.}$$

В этой задаче всегда $t = 239$, $M = 10^9 + 7$.

Принимающая сторона получает, возможно искаженное, сообщение и полиномиальный хеш, рассчитанный для отправленного сообщения. Считается, что сам хеш передается без ошибок. Исправление ошибок происходит по методу наибольшего правдоподобия — необходимо найти сообщение, отличающееся от полученного в минимальном числе бит, и имеющее хеш, равный полученному.

Формат входных данных	Первая строка содержит единственное число K ($1 \leq K \leq 10$) — количество переданных сообщений. Каждая из следующих K строк содержит описание переданного сообщения — три целых числа n ($1 \leq n \leq 34$) — длина сообщения, m ($0 \leq m < 2^n$) — число, двоичная запись которого задает битовую строку полученного сообщения и h ($0 \leq h < 10^9 + 7$) — полученный хеш.
-----------------------------	--

Формат выходных данных	Суммарная длина всех переданных сообщений в одном тесте не превосходит 100. Для каждого сообщения в отдельной строке необходимо вывести единственное число «-1», если это сообщение невозможно расшифровать, иначе необходимо сначала вывести число d — минимальное количество ошибок и далее в этой же строке d чисел — номера битов, которые были искажены при передаче. Если возможных ответов несколько, выведите любой.
------------------------------	---

	Входные данные Выходные данные	
Примеры	3 2 2 239 1 0 1 2 2 238	0 1 0 -1