

Задача А. Распродажа!

Имя входного файла: `booksale.in`
Имя выходного файла: `booksale.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Доктор Стрэндж любит читать и изучать новые заклинания. В его любимом книжном магазине «У Дормамму» акция! Но доктор попал во временную петлю, и у него не хватает времени, поэтому все книги он решил заказать с доставкой через интернет.

Доктор планирует купить n различных книг. По акции при покупке $w + q$ и более книг w самых дешевых из них достаются бесплатно. Доставка одного заказа стоит e монет. Герой может заказать набор различных книг, которых нет в других заказах. Количество книг в одном заказе неограниченно.

К сожалению, доктор забыл значение q . Помогите великому магу. Для каждого значения q от 1 до n найдите минимальное количество монет, которое необходимо потратить, чтобы купить все n книг.

Формат входных данных

В первой строке заданы числа n , e и w — количество книг, цена доставки и количество бесплатных книг ($1 \leq w \leq n \leq 3 \cdot 10^5$; $1 \leq e \leq 10^6$).

Далее следует n строк. В каждой записано число t — цена i -ой книги ($1 \leq t \leq 10^6$).

Формат выходных данных

Выведите n чисел — минимальное количество монет, необходимое для покупки всех книг, при значении q от 1 до n .

Примеры

<code>booksale.in</code>	<code>booksale.out</code>
5 1 1 1 2 3 4 5	12 14 15 15 16
6 3 2 2 1 3 2 1 5	13 15 15 15 17 17

Задача В. Доктор Стрэндж и выставка

Имя входного файла: `exhibition.in`
Имя выходного файла: `exhibition.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У доктора Стрэнджа есть сад, в котором в ряд выставлены n горшков с цветами. На каждом горшке написано некоторое число. На позиции номер i стоит горшок с числом a_i . Иначе говоря, горшки образуют массив a .

Грядет выставка цветов. Доктор Стрэндж выберет для нее ровно k горшков. Он хочет, чтобы его коллекция была самая запоминающаяся. Также доктор Стрэндж любит закономерности, поэтому он верит, что если побитовый AND чисел, написанных на выбранных горшках, будет равняться нулю, то его цветы произведут на всех неизгладимое впечатление.

Помогите доктору Стрэнджу понять, можно ли выбрать k горшков, которые удовлетворяют этому условию.

Побитовый AND — это бинарная операция, действие которой эквивалентно применению логического AND к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов.

Формат входных данных

В первой строке находятся два натуральных числа n, k ($1 \leq n \leq 2 \cdot 10^4, 1 \leq k \leq n$).

В следующей строке находятся n неотрицательных целых чисел a_i ($0 \leq a_i < 2^{12}$).

Формат выходных данных

В первой строке выведите YES, если существует способ выбрать k горшков, чтобы их побитовый AND был равен нулю.

Если ответа не существует — выведите NO.

Примеры

<code>exhibition.in</code>	<code>exhibition.out</code>
3 1 5 4 3	NO
3 2 5 4 3	YES
6 3 6 12 7 8 5 13	YES

Задача С. Граненые стаканы

Имя входного файла: `glasses.in`
Имя выходного файла: `glasses.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Перед доктором Стрэнджем на столе стоит несколько граненых стаканов. Каждый из них при взгляде сверху выглядит как выпуклый многоугольник, а его стенки вертикальны и перпендикулярны столу.

Перед медитацией Стрэндж хочет распределить по стаканам некоторый объем воды, причем так, чтобы высота воды во всех стаканах была одинакова. Гарантируется, что стаканы достаточно высокие, чтобы так можно было сделать.

Помогите ему определить, какой уровень воды должен быть во всех стаканах.

Формат входных данных

В первой строке находятся два целых числа n и v — количество стаканов и объем воды ($1 \leq n \leq 10^5$; $1 \leq v \leq 10^{18}$).

Далее содержится описание стаканов. Форма стакана описывается многоугольником, лежащим в основании. Описание i -го многоугольника начинается со строки, в которой находится единственное число k_i — количество углов в многоугольнике ($3 \leq k_i \leq 10^4$). Далее следует k_i строк, в каждой из которых находится по два целых числа x_{ij} и y_{ij} — координаты очередного угла многоугольника ($|x_{ij}|, |y_{ij}| \leq 10^6$). Углы даны в порядке обхода против часовой стрелки. Суммарное количество углов во всех многоугольниках не превышает 10^6 .

Так как многоугольники описывают только форму стаканов, но не их положение на столе, данные многоугольники могут пересекаться.

Формат выходных данных

В единственной строке выведите одно число — высоту уровня воды в стаканах, с абсолютной или относительной погрешностью не менее 10^{-6} .

Пример

<code>glasses.in</code>	<code>glasses.out</code>
2 10	5.00000000
3	
0 0	
1 0	
0 1	
4	
1 1	
0 2	
0 0	
1 0	

Задача D. Преследование

Имя входного файла: `kdivision.in`
Имя выходного файла: `kdivision.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Кецилий украл жезл Ватумба, и теперь Доктор Стрэндж преследует его по разным альтернативным реальностям. Буквально только что Доктор потерял его, но благодаря своим способностям он все еще может следить, где Кецилий сейчас находится.

Стрэндж записал все номера реальностей, в которых успел побывать Кецилий, и решил найти какую-нибудь закономерность его действий. Однако внезапно Доктор понял, что все номера он записал подряд без пропусков, и теперь невозможно точно восстановить номера реальностей.

Тем не менее Стивен запомнил три важных условия:

- Кецилий побывал ровно в k реальностях.
- Каждая из них имеет неотрицательный номер без лидирующих нулей (кроме реальности с номером 0).
- Из реальности номер x Кецилий всегда попадал в реальность с номером y таким, что абсолютная разность чисел x и y не меньше l , но в то же время не превосходит r .

Точно восстановить передвижения злодея невозможно, поэтому сейчас Стрэндж хочет хотя бы знать количество способов разбить получившееся число на k других, чтобы выполнялись условия, описанные выше.

Формат входных данных

В первой строке содержится число t — количество тестов ($1 \leq t \leq 100$). В i -й из следующих t строк содержится описание i -го теста.

Тест описывается четырьмя числами x, l, r, k — числом, полученным Доктором Стрэнджем, границами для абсолютной разницы и количеством реальностей, в которых побывал Кецилий ($1 \leq x \leq 10^{18}, 0 \leq l \leq r \leq 10^{18}, 1 \leq k \leq 18$).

Формат выходных данных

На каждый тест в отдельной строке выведите ответ на него — искомое количество разбиений.

Пример

<code>kdivision.in</code>	<code>kdivision.out</code>
3	1
248 16 45 2	2
248 16 46 2	0
4444 1 5 2	

Замечание

В первом тесте возможно только одно разбиение, удовлетворяющее ограничениям на абсолютную разность: сначала Кецилий мог быть в реальности номер 24, а потом в реальности номер 8.

Во втором тесте возможны два разбиения: $2 \rightarrow 48, 24 \rightarrow 8$.

Задача Е. Библиотека

Имя входного файла: `library.in`
Имя выходного файла: `library.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Одним из источников знаний доктора Стрэнджа являются книги, которые он довольно часто берет в библиотеке. Все книги, которые Стрэндж берет в библиотеке, он обязательно дочитывает до конца, при этом не забывая сдать каждую из них в срок.

Однако в определенный момент количество книг, взятых доктором в библиотеке, стало быстро расти, и он не успевал следить за сроками сдачи, и в добавок к этому изменились правила сдачи книг в библиотеку. Теперь в день можно сдавать не более одной книги.

Стрэндж очень ценит свое время, и поэтому просит вас помочь ему определить, сможет ли он прочесть все взятые в библиотеке книги и сдать их в срок по новым правилам.

Книги Стрэндж умеет читать параллельно, то есть каждый день он одновременно может читать сколько угодно книг. Книга также может быть сдана и в день прочтения, и в день сдачи.

Формат входных данных

В первой строке входного файла дано число n — количество взятых в библиотеке книг ($1 \leq n \leq 10^5$).

Далее следуют n строк. Каждая i -ая строка содержит три целых положительных числа s_i, f_i, c_i — день, в который доктор взял книгу в библиотеке, день, по прошествии которого книга должна быть сдана, и количество дней, которое требуется доктору для прочтения книги, соответственно ($1 \leq s_i, f_i, c_i \leq 10^9$).

Формат выходных данных

Если доктор Стрэндж сможет сдать все прочитанные им книги в срок, в первой строке выходного файла выведите YES. Иначе, если не сможет, выведите NO.

Примеры

<code>library.in</code>	<code>library.out</code>
3 2 7 3 3 8 1 1 4 2	YES
3 1 3 1 1 3 1 1 3 1	NO

Замечание

В первом тесте из условия доктор может, например, сдать первую книгу в 5-ый день, вторую - в 6-ой, а третью - в 3-ий день.

Во втором тесте доктор не сможет сдать все три книги в срок, так как каждую из них ему нужно сдать во 2-ой или в 3-ий день, что невозможно, так как в день можно сдавать не более одной книги.

Задача F. Магические сферы

Имя входного файла: `magic.in`
Имя выходного файла: `magic.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Доктор Стрэндж владеет оком Агамотто, плащом левитации, а также n магическими сферами. Между некоторыми сферами существует фантомная связь. Доктору необходимо зарядить каждую сферу темной или светлой магией для восстановления баланса сил.

Всего существует m пар связанных сфер, при этом каждая связь обладает определенной силой e . Пусть r — это сумма по всем силам связи. Баланс сил считается успешно восстановленным, если суммарная сила связи между сферами, заряженными одним типом магии, не больше, чем $r/2$.

Помогите Стрэнджу найти правильный тип магии для каждой сферы.

Формат входных данных

В первой строке заданы числа n и m — количество сфер и количество связей ($1 \leq n, m \leq 10^5$).

Далее следуют m строк. В каждой записаны числа a , b и e — связь между сферой a и b с силой равной e ($1 \leq a, b \leq n$; $1 \leq e \leq 10^9$).

Гарантируется, что сферы не связаны сами с собой и не существует более одной связи между каждой парой сфер.

Формат выходных данных

Если ответ существует, требуется вывести n чисел, где i -ое число равно 1, если сфера i заряжена темной магией, либо i -ое число равно 0, если сфера i заряжена светлой магией. Если ответа не существует, выведите `None`.

Примеры

<code>magic.in</code>	<code>magic.out</code>
3 2 1 2 2 2 3 3	0 0 1
5 6 1 2 1 2 3 1 3 4 5 5 1 2 2 5 2 3 1 3	0 1 1 0 0

Замечание

В первом примере значение $r = 5$ и сумма ребер одного цвета равна 2. Раскраска подходит, потому что сумма ребер одного цвета 2 меньше, чем половина суммы всех ребер 2.5.

Задача G. Знания — сила

Имя входного файла: `strength.in`
Имя выходного файла: `strength.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Доктор Стрэндж активно изучает магию. Сегодня он наконец осознал, как распространяются темные силы. Оказывается, они распространяются с помощью так называемых «носителей силы», носителями могут быть кто угодно — люди, предметы, растения. А также каждый характеризуется своим «уровнем» — количеством новых носителей, которых он может породить. Распространение происходит по следующему незамысловатому закону:

- Изначально имеется n носителей, все имеют уровень 1.
- Каждый следующий день носитель уровня i порождает новые i носителей первого уровня, которые становятся активны только на следующий день.
- Сам же носитель переходит на новый уровень $i + 1$ (это означает, что на следующий день он породит уже $i + 1$ новых носителей) и его деятельность на текущий день прекращается.

Всего в распоряжении Стрэнджа имеется k дней. Его интересует, сколько всего носителей появится за это время. За помощью он обратился именно к вам.

Формат входных данных

В единственной строке входного файла содержится два натуральных числа n и k — количество носителей изначально и дней соответственно ($1 \leq n \leq 1000, 1 \leq k \leq 10^5$).

Формат выходных данных

Выведите одно число — ответ на задачу. Так как ответ может получиться слишком большим, выведите его по модулю $10^9 + 7$.

Пример

<code>strength.in</code>	<code>strength.out</code>
3 2	15

Замечание

В таблице приведены данные по носителям в конце каждого дня:

день	первый уровень	второй уровень	третий уровень	всего носителей
0	3	0	0	3
1	3	3	0	2×3
2	$2 \times 3 + 3$	3	3	5×3

Задача Н. Путешествие сквозь миры

Имя входного файла: `travelling.in`
Имя выходного файла: `travelling.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Доктор Стрэндж совсем недавно обрел свои способности, но сразу же решил все структурировать. Получив доступ к бесконечному, но конечно же счетному количеству альтернативных реальностей, он решил их все занумеровать. Сделав это, он посмотрел на проделанную работу и заметил, что его нумерация обладает интересным свойством: все его самые любимые альтернативные реальности имеют номера, состоящие из одинаковых цифр и наоборот — любое число, состоящее из одинаковых цифр, соответствует одной из любимых реальностей Стрэнджа.

Разумеется, смотреть на все реальности сразу у доктора Стрэнджа возможности нет, поэтому в данный момент он смотрит на реальности с номерами $l, l + 1, l + 2, \dots, r$. Разумеется, его заинтересовал вопрос — а сколько различных любимых реальностей он сейчас видит? Дел у него много и времени считать это число самому у него нет, поэтому он попросил вас помочь ему с этой задачей.

Формат входных данных

В первой и единственной строке содержатся два числа l, r — номера крайних альтернативных реальностей, которые видит Доктор Стрэндж ($1 \leq l \leq r \leq 10^{18}$).

Формат выходных данных

В единственной строке выведите количество любимых альтернативных реальностей Стрэнджа, которые он сейчас видит.

Примеры

<code>travelling.in</code>	<code>travelling.out</code>
4 7	4
10 100	9

Замечание

В первом примере все номера состоят из одной цифры, а следовательно соответствуют любимым реальностям Стрэнджа.

Во втором примере подходящие номера — 11, 22, 33, 44, 55, 66, 77, 88, 99, которых ровно 9 штук.

Задача I. Карточный трюк

Имя входного файла: `trick.in`
Имя выходного файла: `trick.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В детстве доктор Стрэндж увидел в передаче по телевизору замечательный карточный трюк, который ему очень хорошо запомнился.

Фокусник брал две карты размера $a \times b$ и $c \times d$ сантиметров. Далее он повторял следующее: располагал первую карту так, чтобы ее длинная сторона была горизонтальна, а вторую так, чтобы ее длинная сторона была вертикальна. Накладывал одну карту на другую так, чтобы один угол совпал. После этого та часть карт, которая была наложена, магическим образом исчезала. После этого он начинал с начала.

Так совпало, что в конце обе карты приняли форму квадратов, их размеры совпали, и после наложения они исчезли. Это и поразило маленького Стрэнджа больше всего.

Стрэндж не уверен, что правильно запомнил размеры карт, поэтому просит вас помочь проверить, мог ли фокус получиться с такими размерами, и если да, то сколько раз нужно будет наложить карты.

Формат входных данных

В единственной строке находятся 4 целых числа: a , b , c и d — размеры исходных карт ($1 \leq a, b, c, d \leq 10^{18}$).

Формат выходных данных

В первой строке выведите «YES», если фокус мог получиться, и «NO» иначе.

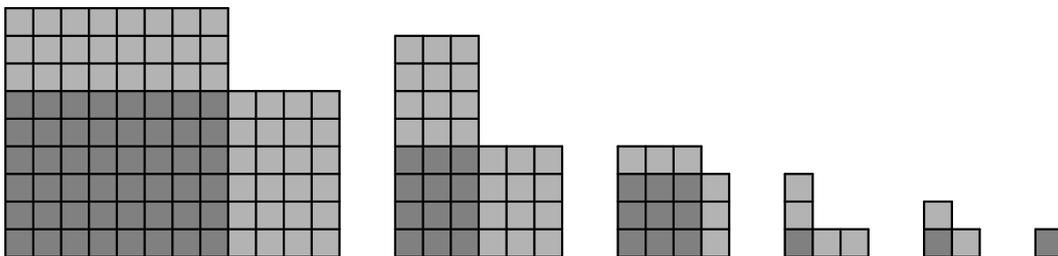
Если фокус мог получиться, во второй строке выведите количество раз, которое нужно наложить карты, прежде чем они исчезнут.

Примеры

<code>trick.in</code>	<code>trick.out</code>
12 6 8 9	YES 6
3 2 1 4	NO

Замечание

Пояснение к первому тесту



Задача J. Две карты

Имя входного файла: `twosegments.in`
Имя выходного файла: `twosegments.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Доктор Стрендж собирается в Тибет, чтобы найти замок Старейшины. Без карты ему не обойтись.

Тибет расположен на прямой. У Доктора есть множество карт, на каждой из которых изображён некоторый отрезок этой прямой. Доктор Стрендж собирается взять с собой ровно две карты, при этом длина той области Тибета, которая изображена на этих картах, должна быть равна s . Обратите внимание, что эта область не обязана быть связной, а если какая-то часть Тибета изображена на обеих картах, её нужно считать только один раз.

Прежде, чем отправиться в путешествие, Доктор приобретает новые карты и продаёт старые.

После каждого изменения коллекции карт он хочет узнать, сколько существует способов выбрать две карты, как описано выше.

Формат входных данных

В первой строке входного файла заданы целые числа s и n — длина, которая интересует Доктора, и количество изменений коллекции карт ($1 \leq s \leq 10^9$, $1 \leq n \leq 10^5$).

В следующих n строках описаны события, происходящие с коллекцией карт:

- Строка вида $1 \ l_i \ r_i$ означает, что Доктор приобрёл новую карту, на которой изображена область Тибета от l_i до r_i ($l_i < r_i$).
- Строка вида $2 \ k$ означает, что Доктор продал карту, которая описывается в событии номер k . События нумеруются с единицы. Гарантируется, что событие номер k — это событие типа 1. Ни одна карта не может быть продана больше одного раза.

Все координаты — целые числа, по модулю не превосходящие $5 \cdot 10^8$. Карты могут быть одинаковыми. Изначально в коллекции Доктора карт нет.

Формат выходных данных

Для каждого события выведите количество способов выбрать две карты так, чтобы длина области Тибета, которая изображена на них, была равна s .

Пример

<code>twosegments.in</code>	<code>twosegments.out</code>
10 6	0
1 0 8	1
1 7 10	2
1 5 15	0
2 2	2
1 12 14	0
2 5	