

Задача А. Диагностика сервиса

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

Вы разрабатываете сервис для управления расписанием авиаперевозок. Для оптимизации сервиса вас попросили разработать библиотеку для диагностики производительности сервиса.

Назовем **span**-ом — непрерывный отрезок времени, который характеризуется следующими параметрами:

- id — идентификатор **span**-а.
- p_{id} — идентификатор родительского **span**-а. Если у **span**-а нет родителя, то считается, что $p_{id} = 0$.
- t_{start} — время начала **span**-а, когда в **span** впервые был осуществлён вход.
- t_{end} — время завершения **span**-а, когда в последний раз был осуществлён выход из **span**-а.

Библиотека должна предоставить следующий интерфейс (набор доступных действий):

- **new id** — создать новый **span** с идентификатором id . Родительским **span**-ом будет считаться последний **span**, в который был осуществлён вход, но ещё не был покинут.
- **drop id** — удалить **span** с идентификатором id . В такой **span** больше нельзя осуществить вход, однако из него по-прежнему можно выйти, а также создать новый дочерний **span**.
- **enter id** — осуществить вход в **span** с идентификатором id .
- **exit** — покинуть текущий **span**. Текущим **span**-ом становится последний **span**, в который был осуществлён вход, но ещё не был покинут.

Для тестирования библиотеки вам предоставили список действий, каждое из которых необходимо выполнить в определённый момент времени. На основе этих данных вам необходимо вычислить характеристики для каждого **span**, в который был осуществлён вход хотя бы раз: идентификатор родителя, время начала и время завершения.

Обратите внимание, что из **span**-а можно осуществить выход до того, как был осуществлён выход из дочерних **span**-ов. В этом случае время выхода должно считаться равным максимальному времени выхода из дочерних **span**-ов. Например, следующая последовательность действий считается корректной:

```
new 1 // create a new span with id = 1
enter 1 // enter the span with id = 1
new 2 // create a new span with id = 2, whose parent is the span with id = 1
exit // exit from the span with id = 1 at time t1
enter 2 // enter the span with id = 2
exit // exit from the span with id = 2 at time t2
drop 1 // delete the span with id = 1
drop 2 // delete the span with id = 2
```

В данном случае время выхода из **span** с $id = 1$ будет считаться равным времени выхода t_2 из **span** с $id = 2$.

Формат входных данных

В первой строке задано единственное число t — количество тестовых примеров. Далее идут t тестовых примеров.

В первой строке тестового примера задано единственное число n — количество событий.

В следующих n строках идут описания действий в одном из форматов:

- t_i new id
- t_i drop id
- t_i enter id
- t_i exit

Гарантируется, что действия идут строго в порядке возрастания t_i . Также предоставляются следующие гарантии:

- При создании нового **span**-а используется ранее неиспользованный id .
- Удаление каждого **span**-а происходит ровно один раз в любой момент после создания **span**-а.
- Вход в **span** осуществляется только после его создания и до его удаления.
- В каждый **span** количество входов совпадает с количеством выходов.
- Выход из **span**-а не может быть осуществлён при отсутствии активного **span**-а.

$$1 \leq t \leq 10^5$$

$$1 \leq n \leq 5 \cdot 10^5$$

$$1 \leq id \leq 10^{18}$$

$$1 \leq t_i \leq 10^{18}$$

Гарантируется, что сумма n не превышает $5 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера в первой строке выведите целое число k — количество **span**-ов, в которые был осуществлён вход хотя бы раз.

В следующих k строках выведите по четыре целых числа id , p_{id} , t_{start} , t_{end} — идентификатор, идентификатор родителя, время начала и время завершения **span**-а соответственно. Строки необходимо вывести в порядке возрастания идентификатора.

Пример

стандартный ввод	стандартный вывод
4	0
4	1
1 new 1	1 0 2 4
2 new 2	2
3 drop 1	1 0 16 93
4 drop 2	2 1 42 75
6	1
1 new 1	1 0 2 6
2 enter 1	
3 new 2	
4 exit	
5 drop 1	
6 drop 2	
10	
12 new 1	
16 enter 1	
22 new 2	
42 enter 2	
55 new 3	
60 drop 1	
73 drop 2	
75 exit	
93 exit	
97 drop 3	
6	
1 new 1	
2 enter 1	
3 enter 1	
4 drop 1	
5 exit	
6 exit	

Задача В. Новая взлетная полоса

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

В последнее время в одном из аэропортов крупной авиакомпании из-за большой загруженности возникают задержки рейсов. Чтобы улучшить качество обслуживания пассажиров, было принято решение построить новую взлетную полосу. Для постройки полосы решили обратиться к вам, как к самому лучшему специалисту.

Территорию аэропорта можно представить как клетчатое поле, левая нижняя клетка которого имеет координаты $(0, 0)$, правая нижняя — координаты $(42 \cdot 10^9, 0)$, левая верхняя — координаты $(0, 42 \cdot 10^9)$, а правая верхняя имеет координаты $(42 \cdot 10^9, 42 \cdot 10^9)$. Вам предоставили n бетонных плит, где i -я плита покрывает прямоугольник ширины w_i и длины h_i клеток. Для успешной постройки взлетной полосы вам необходимо расположить некоторые плиты так, чтобы существовал путь из клетки $(0, 0)$ в клетку (a_x, a_y) , проходящий по соседним клеткам, покрытым бетонными плитами. Клетки считаются соседними, если у них есть общая сторона (левая, правая, нижняя либо верхняя). Бетонные плиты нельзя поворачивать, а также нельзя, чтобы бетонные плиты накладывались друг на друга.

Вам необходимо расположить некоторые из предоставленных плит так, чтобы в результате была получена взлетная полоса.

Формат входных данных

В первой строке задано единственное число t — количество тестовых примеров. Далее идут t тестовых примеров.

В первой строке тестового примера задано три целых числа n , a_x и a_y — количество бетонных плит и координаты клетки соответственно.

В следующих n строках задано по два целых числа w_i и h_i — ширина и длина i -й плиты соответственно.

$$\begin{aligned}1 &\leq t \leq 10^5 \\1 &\leq n \leq 5 \cdot 10^5 \\0 &\leq a_x, a_y \leq 10^9 \\(a_x, a_y) &\neq (0, 0) \\0 &\leq w_i, h_i \leq 10^9\end{aligned}$$

Гарантируется, что сумма n не превышает $5 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера в первой строке выведите «No» (без кавычек) — если невозможно построить плиту из полученных плит. В противном случае в первой строке выведите «Yes» (без кавычек).

Во второй строке выведите число k ($1 \leq k \leq n$) — количество плит, которые необходимо использовать.

В следующих k строках выведите по три целых числа p_i , x_i , y_i ($1 \leq p_i \leq n$, $0 \leq x_i, y_i \leq 10^9$) — номер плиты и координаты левой нижней клетки плиты соответственно. Вам необходимо вывести плиты в таком порядке, чтобы i -я плита и $(i + 1)$ -я плита в выходных данных были соседними. Две плиты считаются соседними, если найдется такая клетка первой плиты и клетка второй плиты, что эти клетки являются соседними.

Каждую плиту можно использовать не более одного раза.

Пример

стандартный ввод	стандартный вывод
1	Yes
3 6 6	3
6 1	1 0 0
1 5	2 6 0
1 2	3 6 5

Задача С. Тёмная сторона интернета

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

Известный видеоблогер Иван решил снять очередное видео про красоты Беларуси и незамедлительно полетел в Минск. Полёт длится целых 2 часа, поэтому, чтобы пассажиры не скучали, авиакомпания подключила интернет прямо на борт самолёта! Иван решил воспользоваться этим функционалом, но так сильно заигрался в видеоигры, что чуть не стал слабоумным! Он твёрдо решил, что пора выбираться из интернета...

Интернет представлен прямоугольной таблицей: её левый нижний угол находится в клетке $(0, 0)$, а верхний правый — в клетке (n, m) . Чтобы выбраться из сети интернет, Ивану необходимо дойти из клетки $(0, 0)$ в клетку (n, m) , переходя только по клеткам, соседним с текущей справа или сверху. Из-за переутомления координата Ивана нарушена, поэтому он пройдет в противоположный угол каким-то из путей равновероятно.

Но не всё так просто! Оказывается, k из этих клеток принадлежат тёмной стороне интернета! Если Иван наступит на такую клетку в первый раз на пути, ему добавят 1 год пребывания в аду! Если наступит на вторую такую клетку — добавят **ещё** целых 2 года! И так далее: за клетку под номером i по счёту на пути ему добавят **ещё** i лет пребывания в аду. Обратите внимание, что если начальная или конечная клетка принадлежит тёмной стороне интернета, Иван наступит на неё в любом случае.

Иван зол от того, что интернет вскружил ему голову, и не помнит, каким именно путём он вышел из интернета. Поэтому вы, как директор авиакомпании, должны посчитать математическое ожидание количества лет, которое он проведёт в аду. Иначе он подаст на вас в суд, ведь до полёта Ивану не пришлось бы проводить в аду ни дня!

Можно показать, что ответ можно представить в виде $\frac{p}{q}$, где p и q — взаимно простые целые числа ($q \not\equiv 0 \pmod{10^9 + 7}$). Обозначим за q^{-1} целое число, для которого верно $q \cdot q^{-1} \equiv 1 \pmod{10^9 + 7}$. Выведите значение $p \cdot q^{-1}$ по модулю $10^9 + 7$.

Формат входных данных

В первой строке вводятся три целых числа: n , m и k .

$$1 \leq n, m \leq 5 \cdot 10^6$$

$$0 \leq k \leq \min(5000, (n + 1) \cdot (m + 1))$$

В каждой из следующих k строк вводятся по два целых числа: x и y — координаты клетки на тёмной стороне интернета (гарантируется, что каждая клетка во входных данных уникальна).

$$0 \leq x \leq n$$

$$0 \leq y \leq m$$

Формат выходных данных

Выведите единственное целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 5 5 0 4 1 3 2 2 3 1 4 0	1
3 3 2 1 1 2 2	200000003
10 10 5 0 0 4 7 2 8 6 9 10 10	599184879

Замечание

В первом тестовом примере каждый путь пройдёт ровно через одну клетку тёмной стороны интернета, поэтому математическое ожидание количества лет в аду равно 1.

Во втором тестовом примере можно посчитать, что 4 пути пройдут через 0, 8 путей пройдут через 1, а 8 путей через 2 клетки тёмной стороны интернета, поэтому математическое ожидание количества лет в аду равно $\frac{4 \cdot 0 + 8 \cdot 1 + 8 \cdot (1+2)}{4+8+8} = \frac{32}{20} = \frac{8}{5}$.

Задача D. Ключ от сейфа

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

Иван очень любит путешествовать, но еще больше он любит привозить различные памятные сувениры из стран, где он побывал. Чтобы привезти сувенир в целости и сохранности, Иван использует свой любимый сейф, который он купил в одном из путешествий.

Замок от сейфа — это состоящий из k ячеек кодовый механический замок, где каждая ячейка представляет собой циклический механизм для выбора ровно одной цифры. Для открытия замка необходимо ввести комбинацию, которая использовалась для закрытия замка.

За долгое время путешествий Иван выработал особую стратегию, по которой он выбирает комбинации для закрытия своего сейфа. Комбинация s считается валидной тогда и только тогда, если её можно разделить на множество уникальных подстрок, которые представляют собой строки Туэ-Морса для алфавита $\{0, 1\}$.

Последовательность строк Туэ-Морса для алфавита $\{0, 1\}$ — это последовательность строк, которая состоит только из символов $\{0, 1\}$ и генерируется по следующему правилу:

$$t_0 = 0,$$
$$t_k = \text{concat}(t_{k-1}, \text{inverse}(t_{k-1})),$$

где $\text{concat}(x, y)$ — это конкатенация строк x и y , а $\text{inverse}(s)$ — это инверсия символов строки s на алфавите $\{0, 1\}$, то есть 0 меняется на 1, а 1 меняется на 0.

Например, первые 5 строк последовательности Туэ-Морса выглядят следующим образом:

$$t_0 = 0$$
$$t_1 = 01$$
$$t_2 = 0110$$
$$t_3 = 01101001$$
$$t_4 = 0110100110010110$$

Иван разбирается в безопасности и имеет определенный уровень паранойи, поэтому после каждого открытия сейфа Иван может поменять текущую комбинацию на сейфе на любую последовательность символов (необязательно валидную), чтобы никто не подсмотрел предыдущую комбинацию, которой он закрывал сейф.

Так как Иван разбирается не только в безопасности, но и в оптимизации, то каждый раз, когда он закрывает сейф, он хочет сделать минимальное количество прокруток ячеек замка.

Вам дана строка s — текущая комбинация символов на замке сейфа, вам необходимо определить, какое минимальное количество оборотов ячеек замка необходимо сделать, чтобы из s получить валидную комбинацию.

Формат входных данных

Единственная строка s — начальная комбинация на замке от сейфа длины $k \leq 10^6$.

Формат выходных данных

Единственная строка, которая содержит минимальное количество операций для превращения строки s в валидную комбинацию.

Примеры

стандартный ввод	стандартный вывод
8009	5
90091001	4
0110	0

Замечание

Пояснение к примеру 8009. Минимальное количество операций 5 достигается следующим образом:

- $8 \rightarrow 9 \rightarrow 0$ (2 операции)
- $0 \rightarrow 1$ (1 операция)
- $0 \rightarrow 1$ (1 операция)
- $9 \rightarrow 0$ (1 операция)

Задача Е. Обслуживание

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

К огромному сожалению, самолеты могут ломаться, поэтому их нужно постоянно обслуживать. Вам дано m самолетов, каждый из которых имеет свой размер b_i . Также у вас имеется n ангаров, каждый из которых так же имеет свой размер a_i . Чтобы обслужить b_i самолет в a_j ангаре, потребуется $\frac{b_i}{a_j}$ часов, так же нужно учесть, что если размер самолета превышает размер ангара, то обслужить данный самолет не получится.

В данный момент все самолеты стоят в ряд напротив ангаров, поэтому в связи с тем, что перемещать их очень трудно, самолеты нужно разделить на n групп, где в каждой группе будет находиться некоторое число подряд идущих самолетов. Количество самолетов в группе может быть нулевым. Менять самолеты местами нельзя. Пусть группа с номером i начинается с l_i самолета и заканчивается r_i самолетом, для всех групп $1 \leq i, j \leq n$ и $i < j$ должно выполняться одно из двух условий:

- $r_i < l_j$.
- В группе i или j нет самолетов.

Все самолеты должны быть выбраны в группу. После того, как самолеты разделены на группы, в i ангаре по очереди будут обслуживаться самолеты из i группы. Каждый из ангаров обслуживает самолеты одновременно. Ваша задача узнать, за какое минимальное количество часов выполнится обслуживание самолетов, если же обслужить самолеты не получится, выведите -1.

Формат входных данных

В первой строке вводится целое число n — количество ангаров.

Во второй строке вводится n целых чисел a_i — размер ангара под номером i .

В третьей строке вводится целое число m — количество самолетов.

В четвертой строке вводится m целых чисел b_i — размер самолета под номером i .

$$1 \leq n, m \leq 10^5$$

$$1 \leq a_i, b_i \leq 10^5$$

$$1 \leq b_i \leq 10^5$$

Формат выходных данных

Выведите одно число, минимальное количество часов, если же обслужить самолеты невозможно, выведите -1. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 2 6 4 4 1 2 6 4	1.33333333
2 1 3 3 1 1 4	-1

Задача F. Отпуск

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

О нет! На борту самолета произошел перевес, и теперь отпуск Вадима на Бали под угрозой.

Самолет, который должен вот вот вылететь, вмещает f литров топлива. Известно, что самолет потребляет $\frac{v}{m} + c$ литров топлива на 100 километров, где v — скорость самолета, m — максимально возможная скорость самолета и c — постоянное минимальное потребление топлива на холостых оборотах. Вадим изучил карту и понял, что лететь ему нужно ровно s километров. Теперь Вадим интересуется, можно ли выбрать такую скорость полета, что он сможет добраться до пункта назначения. Можно считать, что самолет набирает любую скорость в пределах $(0, m]$ моментально.

Формат входных данных

В первой строке задано четыре целых числа f, m, c, s — объем топливного бака, максимально допустимая скорость, постоянное минимальное потребление топлива и расстояние, которое необходимо преодолеть.

$$1 \leq f, m, c, s \leq 10^6$$

Формат выходных данных

В единственной строке выведите «Yes» (без кавычек), если Вадим сможет долететь до Бали, и «No» (без кавычек) в противном случае.

Примеры

стандартный ввод	стандартный вывод
450 1500 15 3000	No
451 1500 15 3000	Yes

Задача G. Правильные маршруты

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

Вадим недавно узнал про правильные скобочные последовательности. Правильной скобочной последовательностью называется такая строка, которую можно получить, используя следующие правила:

- $()$ — правильная скобочная последовательность.
- (x) — правильная скобочная последовательность, если x также правильная скобочная последовательность.
- xy — правильная скобочная последовательность, если x и y также правильные скобочные последовательности.

Примеры правильных скобочных последовательностей: $((()()))$, $()()()()$, $()$.

Примеры строк, не являющихся правильной скобочной последовательностью: $((()$, $)()$.

У Вадима есть карта, состоящая из n городов. Города соединены $n - 1$ авиарейсом, каждый из которых позволяет перемещаться в любом направлении. Гарантируется, что из любого города можно добраться до любого другого с использованием одного или нескольких авиарейсов. Вадим решил выписать m строк, состоящих только из символов « $($ » и « $)$ », и назначить каждому авиарейсу одну из этих строк (строки могут повторяться).

Теперь Вадим интересуется, а сколько существует упорядоченных пар городов (a, b) ($a \neq b$) таких, что маршрут между этими городами является правильным. Маршрут считается правильным, если полученная конкатенацией строк авиарейсов на пути из a в b является правильной скобочной последовательностью.

Формат входных данных

В первой строке задано единственное число t — количество тестовых примеров. Далее идут t тестовых примеров.

В первой строке тестового примера задано два целых числа n и m — количество городов и количество строк соответственно.

В следующих m строках задано по одной строке s_i , состоящей только из символов « $($ » и « $)$ ».

В следующих $n - 1$ строках задано по три целых числа a_i , b_i , j_i — номера городов, соединенных авиарейсом, и номер строки соответственно.

$$1 \leq t \leq 10^5$$

$$2 \leq n \leq 2 \cdot 10^5$$

$$1 \leq m < n$$

$$|s_i| \geq 1$$

$$1 \leq a_i, b_i \leq n$$

$$a_i \neq b_i$$

$$1 \leq j_i \leq m$$

Гарантируется, что сумма n не превышает $2 \cdot 10^5$, а сумма длин всех строк s_i не превышает $5 \cdot 10^5$.

Пример

стандартный ввод	стандартный вывод
3	9
5 3	3
((2
))	
(())	
1 2 1	
2 3 2	
3 4 3	
4 5 3	
5 2	
(
)	
1 2 1	
1 3 2	
3 4 1	
3 5 2	
2 1	
()	
1 2 1	

Замечание

Во втором тестовом примере подходят следующие пути: $[2, 1, 3]$, $[4, 3, 5]$ и $[4, 3, 1]$.

В третьем тестовом примере подходят следующие пути: $[1, 2]$ и $[2, 1]$.

Задача Н. Эксперимент с зеркалами

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

В оптической лаборатории создана система из n строк и m столбцов. В некоторых ячейках установлены k зеркал под углом в 45 градусов к вертикальной или горизонтальной прямой: «/» и «\». Лазерный луч стартует слева из позиции $(r, 0)$ для строки r и движется вправо. Строка считается безопасной, если луч достигает правой границы $c = m + 1$, и опасной — если заикликивается или выходит через другую границу.

Вас просят провести q экспериментов. Для каждого эксперимента вам нужно определить, сколько строк в диапазоне $[a, b]$ считаются безопасными.

Формат входных данных

В первой строке задано единственное число t — количество тестовых примеров. Далее идут t тестовых примеров.

В первой строке тестового примера задано три целых числа n , m и k — число строк и столбцов в оптической системе, а также количество установленных зеркал.

В следующих k строках задано по два целых числа и одному символу r_i, c_i, t_i ($t \in \{/, \backslash\}$) — номер строки и столбца, где расположено зеркало, а также его направление (символ «/» или «\»).

В следующей строке задано единственное целое число q — количество экспериментов.

В следующих q строках задано по два целых числа a_i и b_i — диапазон строк для i -го эксперимента.

$$1 \leq t \leq 10^5$$

$$1 \leq n, m \leq 10^9$$

$$0 \leq k \leq 2 \cdot 10^5$$

$$1 \leq q \leq 2 \cdot 10^5$$

$$1 \leq r_i \leq n$$

$$1 \leq c_i \leq m$$

$$1 \leq a_i \leq b_i \leq n$$

Гарантируется, что сумма k и сумма q не превышают $2 \cdot 10^5$.

Формат выходных данных

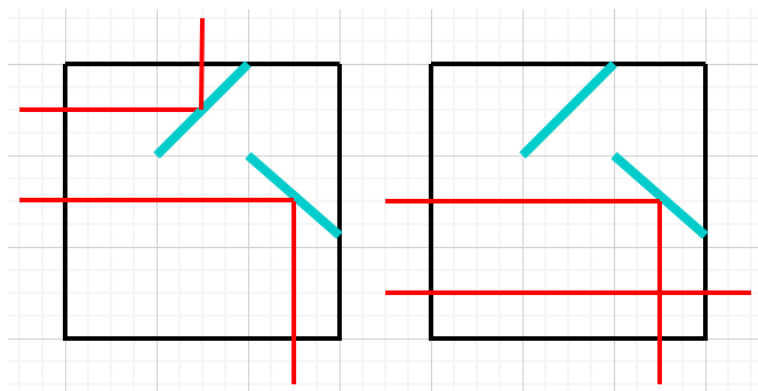
Для каждого тестового примера выведите q строк. В i -й строке выведите единственное целое число — количество безопасных строк для i -го эксперимента.

Пример

стандартный ввод	стандартный вывод
2	1
10 10 5	1
1 1 \	3
4 1 \	0
5 3 /	1
3 3 /	
6 6 /	
3	
1 1	
2 4	
1 6	
3 3 2	
1 2 /	
2 3 \	
2	
1 2	
2 3	

Замечание

Рассмотрим второй тестовый пример. Для первого эксперимента был направлен один луч в первой строке, который был отражен вверх, и второй луч, который был отражен вниз. Оба луча не добрались до правой границы. Во втором эксперименте луч во второй строке был отражен вниз, а луч в третьей строке беспрепятственно добрался до правой границы.



Задача I. Палиндром

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

Авиакомпания «Chorik Airways» использует необычную систему нумерации рейсов. Каждый рейс имеет уникальный номер, который состоит из двух чисел — $number$ и $base$. Самое крутое то, что если мы представим число $number$ в системе счисления $base$, это число окажется палиндромом чётной длины.

Палиндромом называется число, читающееся одинаково слева направо и справа налево. Например, 101101 — палиндром, а 1010 — нет. Под длиной числа подразумевается количество цифр в нём.

У каждого самолёта есть три параметра — количество пассажиров a , грузоподъёмность b и дальность полёта c . Их произведение задаёт число $number$, которое используется в номере самолёта. Но так как у нескольких самолётов значения $a \cdot b \cdot c$ могут совпадать, могут быть сложности с поиском нескольких подходящих $base$. Вам, как главному программисту компании «Chorik Airways», поручена задача — найти количество подходящих $base$ для заданных a, b, c .

Можно показать, что количество таких $base$ — конечное число.

Формат входных данных

В первой строке вводятся три целых числа: a, b и c .

$$1 \leq a, b, c \leq 10^6$$

Формат выходных данных

Выведите единственное целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1 3 3	2
1 3 5	3
52 42 69	48

Замечание

В первом тестовом примере число $number$ равно 9. Можно показать, что подходят только $base = 2$ ($9_{10} = 1001_2$) и $base = 8$ ($9_{10} = 11_8$).

Задача J. Это, походу, бомба!

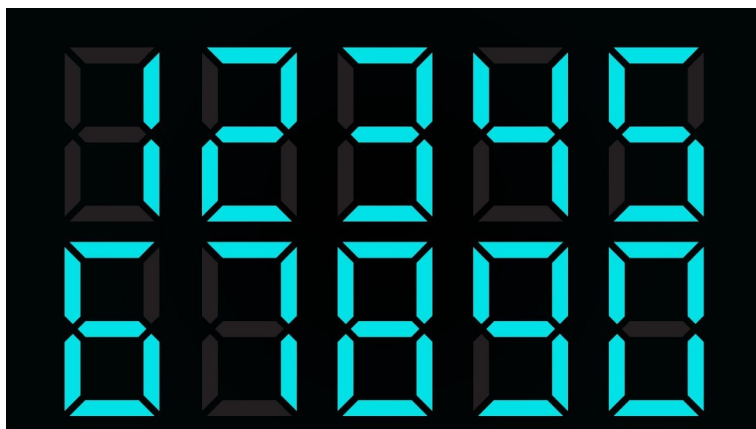
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На борту есть сапёр? Не переживайте, я просто спросил...

Злоумышленник пронёс бомбу прямо на борт самолёта! Благо, экипаж заблаговременно обнаружил это, а вы оказались чемпионом мира по игре в сапёр, поэтому вам не должно составить труда обезвредить бомбу.

Но бомба оказалась непростой: её таймер состоит из n разрядов, а некоторые сегменты на таймере бомбы сломаны, то есть не горят никогда. После расспроса злоумышленник рассказал, какие сегменты не горят в первых m разрядах, а для каждого следующего разряда не горят ровно те сегменты, что не горят у разряда на m левее. Также, если у оставшегося времени меньше цифр, чем разрядов в таймере, то будут отображаться ведущие нули.

Цифры на таймере записываются следующим образом:



После тщательного анализа таймера вы обнаружили, что отсчёт ещё не начался. Кроме того, если обозначить цифру в разряде i за $digit_i$, то $digit_i = digit_{i-1} \cdot a + b \bmod 10$ для всех $i > m$.

Но внезапно таймер начал свой отсчёт! После очередной беседы со злоумышленником он признался: чтобы обезвредить бомбу, необходимо найти количество значений таймера, меньших или равных начальному, в которых не сломан ни один горящий сегмент.

Так как ответ может быть очень большим, его нужно взять по модулю $10^9 + 7$.

Формат входных данных

В первой строке вводится число n — количество разрядов в времени отсчёта.

Во второй строке вводится число m — количество первых известных разрядов времени отсчёта.

В третьей строке вводится два числа a и b — коэффициенты.

В четвёртой строке вводится m первых цифр времени отсчёта без пробелов.

Для каждого i от 1 до m в строке $4 + i$ вводится целое число $0 \leq mask_i < 2^7$ — битовая маска сломанных сегментов. Если j -й бит числа $mask_i$ равен 1, — значит j -й сегмент в i -м разряде не работает. Нумерацию сегментов можно посмотреть на картинке ниже.

$$1 \leq n \leq 10^{18}$$

$$1 \leq m \leq 10^5$$

$$1 \leq a, b \leq 100$$

Формат выходных данных

Одна строка — ответ на задачу по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 1 1 1 5 0	57
1 1 55 89 3 72	0

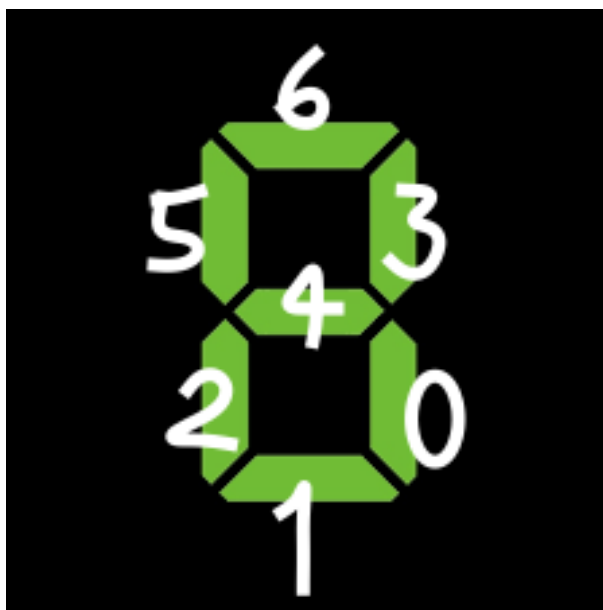
Замечание

В первом тестовом примере мы вычисляем полное значение таймера длиной $n = 2$. Поскольку $m = 1$, мы вычисляем вторую цифру, используя: $digit_2 = 5 * 1 + 1 = 6$

Все сегменты работают, значит мы можем использовать любую цифру от 0 до 9.

Итак, всего допустимых цифр: От 00 до 56, то есть 57 значений.

Нумерация сегментов в цифре:



Задача К. Собеседование

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Для того чтобы попасть на собеседование к одному крупному авиаперевозчику, необходимо решить следующую задачу.

Вам предоставили массив длины $42 \cdot 10^9$, состоящий из целых чисел, изначально заполненный 0-ми.

Вас просят последовательно выполнить n операций, где i -я операция описывается четырьмя целыми числами l_i , r_i , a_i и b_i , которые означают, что для каждого индекса массива j такого, что $l_i \leq j \leq r_i$, значение соответствующего элемента массива необходимо увеличить на $a_i + b_i \cdot (j - l_i)$.

Затем вас просят для каждого элемента массива вычислить остаток от деления на m и полученные остатки сложить. Результат сложения и будет являться ответом на задачу.

Формат входных данных

В первой строке задано единственное число t — количество тестовых примеров. Далее идут t тестовых примеров.

В первой строке тестового примера задано два целых числа n и m — количество операций, которые необходимо выполнить, и число для получения остатков от деления соответственно.

В следующих n строках задано по четыре целых числа l_i , r_i , a_i , b_i — описание операции, которую нужно выполнить над массивом.

$$\begin{aligned} 1 &\leq t \leq 10^5 \\ 1 &\leq n \leq 5 \cdot 10^5 \\ 2 &\leq m \leq 10^9 + 7 \\ 1 &\leq l_i \leq r_i \leq n \\ 0 &\leq a_i, b_i \leq m \end{aligned}$$

Гарантируется, что сумма n не превышает $5 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера выведите единственное целое число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
3	20
2 100	1297176542
4 4 5 10	10
1 3 2 3	
3 1000000007	
1 1000 0 1	
500 5000 42 128	
1 2000 100 0	
2 2	
1 10 0 1	
1 10 1 1	

Замечание

В третьем тестовом примере массив после первой операции выглядел так:

$$[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, \dots]$$

После второй операции массив выглядел так:

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 0, ...]