

Задача А. Редкие праздники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Главный астрономический праздник отмечается 29 февраля. Как известно, 29 февраля бывает только в високосные годы. Определите, сколько раз отмечался праздник между годами L и R включительно.

Год Y считается високосным, если выполняется одно из условий:

- Y делится на 400;
- Y делится на 4, но не делится на 100.

Формат входных данных

В единственной строке вводятся два целых числа L и R ($0 \leq L \leq R \leq 10^{15}$) — границы интересующего отрезка.

Формат выходных данных

Выведите единственное целое число — количество високосных лет в отрезке $[L, R]$.

Система оценки

Решения, верно работающие в случае $L \leq R \leq 10^6$, будут получать не менее 50 баллов.

Примеры

стандартный ввод	стандартный вывод
0 10	3
1999 2026	7

Замечание

В первом примере високосными годами являются годы 0, 4 и 8.

Во втором примере високосными годами являются 2000, 2004, 2008, 2012, 2016, 2020, 2024.

Задача В. Очередь в столовую

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

Женя пришел в столовую и увидел, что в очереди стоит n человек. У каждого человека есть предел терпения t_i — сколько секунд он готов ожидать своей очереди, и время заказа k_i — сколько секунд он делает заказ. Время ожидания человека — это суммарное время заказа для всех людей перед ним. Гарантируется, что изначально время ожидания для каждого человека в очереди не превосходит его предел терпения.

Женя — очень дружелюбный человек, поэтому знает всех, кто стоит в очереди, и может встать в любое место. Он хочет встать как можно ближе к началу очереди, но не хочет, чтобы из-за него чье-то время ожидания превысило предел терпения. При этом у самого Жени терпение бесконечно.

Формат входных данных

В первой строке дано одно целое число $cases$ ($1 \leq cases \leq 10000$) — количество наборов входных данных. Далее следуют описания каждого из наборов входных данных.

В первой строке каждого набора входных данных записано два целых числа n и d ($1 \leq n \leq 10^5$, $1 \leq d \leq 10^9$) — количество людей в очереди и длительность заказа Жени.

В i -й из последующих n строк записаны два целых числа t_i и k_i ($1 \leq t_i, k_i \leq 10^9$) — предел терпения и длительность заказа i -го человека в очереди.

Пусть N равно сумме n по всем наборам входных данных. Гарантируется, что $N \leq 10^6$.

Формат выходных данных

Выведите, на какое минимальное по счёту место может встать Женья.

Система оценки

Всего в этой задаче 20 тестов (не считая теста из условия). Оценка потестовая, каждый из тестов, кроме теста из условия, оценивается в 5 баллов. На некоторые из тестов наложены дополнительные ограничения, указанные в таблице.

Обозначим за N сумму по n по наборам входных данных.

Номер теста	Дополнительные ограничения	Комментарий
1	—	Тест из условия
2, 3	$n \leq 10, N \leq 100$	—
4, 5, 6	Все t_i равны	—
7, 8, 9	$k_i = d = 1$	—
10, 11, 12, 13, 14	$n \leq 1000, N \leq 10\,000$	—
15, 16, 17, 18, 19, 20, 21	—	—

Пример

стандартный ввод	стандартный вывод
2	1
3 2	5
5 1	
6 2	
10 3	
4 3	
2 2	
4 1	
5 3	
8 2	

Замечание

В первом наборе входных данных Женья может встать первым — в самое начало очереди, тогда следующий после него человек, имея предел 5 секунд, прождет 2 секунды, следующий, имея предел 6 секунд, прождет $2 + 1 = 3$ секунды, последний, имея предел 10 секунд, прождет $2 + 1 + 2 = 5$ секунд.

Во втором наборе входных данных, куда бы Женья не встал, чье-то терпение переполнится, поэтому он становится последним.

Задача С. Факультет Дедлайнов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.6 секунд
Ограничение по памяти:	256 мегабайт

Климентий учится на Факультете Дедлайнов. Семестр длится m дней, в течение семестра Климентию предстоит выполнить n заданий.

Задание с номером i нужно выполнить с дня l_i до дня r_i включительно. Задание можно выполнить в любой день p ($l_i \leq p \leq r_i$).

Также для каждого дня $p = 1, 2, \dots, m$ известна продуктивность Климентия, которая выражается целым неотрицательным числом a_p .

Массив значений продуктивности $a_1, a_2, \dots, a_m$ задан следующим рекуррентным соотношением:

- $a_1 = x$
- Для всех $k = 2, 3, \dots, m$ $a_k = (a_{k-1} \cdot b + c) \% mod$, где $\%$ — операция взятия остатка от деления, а mod — число, остаток от деления на которое вычисляется.

Климентий хочет посвятить выполнению заданий только один день в семестре. Необходимо выбрать такой день p , чтобы произведение $cnt(p) \cdot a_p$ было максимально. Здесь $cnt(p)$ — количество заданий, которые можно сдать в день p .

Определите максимальное произведение $cnt(p) \cdot a_p$.

Формат входных данных

В первой строке вводятся два целых числа n, m — количество заданий и количество дней в семестре ($1 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 2 \cdot 10^9$).

Во второй строке вводятся четыре целых числа x, b, c, mod , определяющие массив продуктивностей a ($0 \leq x < mod, 0 \leq b, c \leq 10^9, 1 \leq mod \leq 2 \cdot 10^5$).

В следующих n строках вводятся границы дней для выполнения каждого из n заданий. Каждая граница описывается двумя числами l_i, r_i ($1 \leq l_i \leq r_i \leq m$).

Формат выходных данных

В единственной строке выведите одно целое число — максимальное значение $cnt(p) \cdot a_p$.

Система оценки

Решения, работающие для ограничений $n, m \leq 2000$, наберут не меньше 20 баллов

Решения, работающие для ограничений $m \leq 2 \cdot 10^5$, наберут не меньше 40 баллов

Решения, работающие для ограничений $n, mod \leq 2000$, наберут не меньше 60 баллов

Решения, работающие для ограничений $mod \leq 2000$, наберут не меньше 74 баллов

Решения, работающие без дополнительных ограничений, наберут 100 баллов

Примеры

стандартный ввод	стандартный вывод
3 10 1 2 3 7 1 3 2 6 5 10	12
7 12 3 4 2 9 1 4 3 8 6 12 2 2 9 12 4 10 1 12	32

Замечание

- *Рассмотрим первый пример*

Массив продуктивностей выглядит следующим образом:

1, 5, 6, 1, 5, 6, 1

Для каждого $p = 1, 2, \dots, 10$ количество заданий, которые можно сдать в день p , равно:

1, 2, 2, 1, 2, 2, 1, 1, 1, 1

И для каждого $p = 1, 2, \dots, 10$ произведение $cnt(p) \cdot a_p$ равно:

1, 10, 12, 1, 10, 12, 1, 5, 6, 1

Таким образом, максимальный ответ равен 12.

- *Рассмотрим второй пример*

Массив продуктивностей выглядит следующим образом:

3, 5, 4, 0, 2, 1, 6, 8, 7, 3, 5, 4

Для каждого $p = 1, 2, \dots, 12$ количество заданий, которые можно сдать в день p , равно:

2, 3, 3, 4, 3, 4, 4, 4, 4, 4, 3, 3

И для каждого $p = 1, 2, \dots, 12$ произведение $cnt(p) \cdot a_p$ равно:

6, 15, 12, 0, 6, 4, 24, 32, 28, 12, 15, 12

Таким образом, максимальный ответ равен 32.

Задача D. Хорошие палиндромы

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

Рассмотрим процесс на строке s , состоящей только из символов "a" и "b". На каждом шаге этого процесса из строки одновременно удаляются первое и последнее вхождение букв "a" и "b" (если такие вхождения есть). Обратите внимание, что первое и последнее вхождение могут совпадать. Нас интересует первый момент, после которого строка s станет палиндромом. Палиндромом называется строка, которая читается одинаково слева-направо и справа-налево. Пустая строка считается палиндромом. Номер шага, после которого строка становится палиндромом, назовем красотой строки.

Пусть $s = \text{"aabaababa"}$. Первые и последние вхождения каждой буквы подчеркнуты.

После первого шага процесса $s = \text{"aaaba"}$. Здесь первое и последнее вхождение буквы "b" совпадают.

После второго шага процесса $s = \text{"aa это палиндром, поэтому красота } s \text{ равна } 2$.

Дана строка t , состоящая из символов "a" и "b". Вам поступит q запросов двух типов:

! i — изменить i -й символ в t на другой ("a"меняется на "b"и наоборот).

? $l\ r$ — узнать красоту подстроки $t[l...r]$.

Обратите внимание, что запросы первого типа влияют на все последующие запросы, а запросы второго типа никак не меняют строку t .

Формат входных данных

В первой строке входных данных вводится строка t ($1 \leq |t| \leq 2 \cdot 10^5$). Гарантируется, что t состоит только из букв "a"и "b".

В следующей строке входных данных вводится единственное число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов.

Следующие q строк содержат описания запросов. Описание каждого запроса имеет следующий вид:

! i ($1 \leq i \leq n$) — изменить символ в строке.

? $l\ r$ ($1 \leq l \leq r \leq n$) — узнать красоту подстроки $t[l...r]$.

Формат выходных данных

Для каждого запроса второго типа выведите соответствующую красоту подстроки.

Система оценки

Решения, верно работающие при $|t|, q \leq 100$ будут набирать не менее 12 баллов.

Решения, верно работающие при $t = \text{"ababa..."}$ и $|t|, q \leq 100000$ и без запросов первого типа будут набирать не менее 12 баллов.

Решения, верно работающие при $|t|, q \leq 5000$ будут набирать не менее 28 баллов.

Решения, верно работающие при $|t|, q \leq 100000$ и без запросов первого типа будут набирать не менее 35 баллов.

Решения, верно работающие при $|t|, q \leq 80000$ будут набирать не менее 63 баллов.

Примеры

стандартный ввод	стандартный вывод
aabab	0
4	1
? 3 5	0
? 1 5	
! 1	
? 1 5	
aaababbaba	2
6	3
? 1 10	1
! 2	2
? 1 10	
? 2 7	
! 4	
? 2 8	

Замечание

Разберем первый пример.

Первый запрос приходит к подстроке $t[3...5] = \text{"bab"}$, которая уже является палиндромом, поэтому ответ на этот запрос равен 0.

Второй запрос приходит ко всей строке t , которая палиндромом не является. После одного шага

процесса строка бы превратилась в "а", что уже является палиндромом, поэтому ответ на этот запрос равен 1.

Третий запрос меняет в строке t первый символ, после чего $t = "babab"$.

Четвертый запрос снова приходит ко всей строке, которая стала палиндромом после третьего запроса, поэтому ответ на этот запрос равен 0.

Задача E. Graphforces

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	512 мегабайт

Дан связный граф, в котором n вершин и m двунаправленных ребер. Для каждой вершины $0 \leq v \leq n - 1$ (нумерация вершин начинается с нуля) вам нужно решить следующую подзадачу:

Пусть вершины $u_1, \dots, u_k$ — соседи вершины v . Каждое ребро (v, u_i) требуется заменить на ребро (v, x) , где либо $x = (n + u_i - 1) \% n$, либо $x = (n + u_i + 1) \% n$ (здесь $\%$ — операция взятия остатка от деления). Оригинальное ребро (v, u_i) удаляется из графа, а новое ребро может образовывать петлю или совпадать с уже существующим ребром. Если $u_i = 0$, то новые ребра могут вести в вершину 1 или $n - 1$.

Вам нужно, чтобы количество компонент связности в измененном графе было минимально возможным — это значение и будет ответом для очередной независимой подзадачи. Для каждой вершины v рассматривается отдельно граф G_v , полученный из исходного графа только заменой рёбер, инцидентных v . Замены для разных v не влияют друг на друга.

Компонентой связности называется подмножество вершин графа, между которыми существует путь, при этом к этому подмножеству нельзя добавить ни одной вершины. Компонента связности может состоять и из одной вершины.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m ($2 \leq n \leq 10^6$, $n - 1 \leq m \leq 10^6$) — количество вершин в графе и количество рёбер.

Каждая из следующих m строк содержит два целых числа v и u ($0 \leq v, u < n$) — очередное ребро графа. Гарантируется, что в графе нет петель и кратных рёбер.

Формат выходных данных

Выведите n чисел, где i -тое число является ответом для вершины с номером i .

Система оценки

Все решения должны проходить тесты из условия.

Решения, верно работающие на графах, степень каждой вершины в которых не превосходит 2, будут набирать не менее 17 баллов.

Решения, верно работающие при $n, m \leq 15$, будут набирать не менее 12 баллов.

Решения, верно работающие на деревьях (т.е. при $m = n - 1$), будут набирать не менее 44 баллов.

Решения, верно работающие при $n, m \leq 5000$, будут набирать не менее 32 баллов.

Примеры

стандартный ввод	стандартный вывод
4 3 1 2 2 3 3 0	1 1 2 2
8 8 0 2 2 1 2 6 2 3 0 7 7 4 7 5 4 5	2 1 3 1 1 1 1 1

Замечание

Разберем ответ для вершины 3 в первом тесте. Ребро (3, 2) можно заменить либо на ребро (3, 1), либо на ребро (3, 3). Также и ребро (3, 0) можно заменить либо на ребро (3, 1), либо на ребро (3, 3). При любом выборе вершины 2 и 3 будут лежать в разных компонентах связности, а вершины 3, 0 и 1 будут лежать в одной компоненте связности, поэтому ответ на эту независимую подзадачу — 2.