

## Задача А. Пётя и рекурента

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

Пётя сидел на уроке по линейной алгебре и призадумался над интересной задачей.

Пусть  $f_x = c^{2x-6} \cdot f_{x-1} \cdot f_{x-2} \cdot f_{x-3}$ .

Найдите  $f_n$  по модулю  $10^9 + 7$ .

### Формат входных данных

Единственная строка входных данных содержит пять чисел —  $n, f_1, f_2, f_3, c$  ( $4 \leq n \leq 10^{18}$ ,  $1 \leq f_1, f_2, f_3 \leq 10^9$ ).

### Формат выходных данных

Одно число — ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
18 93 42 342 11	502855435
5 4 3 2 2	9216

## Задача В. Дискретное логарифмирование

Имя входного файла: `log.in`  
Имя выходного файла: `log.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Даны натуральные числа  $a$ ,  $b$ ,  $n$ . Требуется найти *дискретный логарифм*  $b$  по основанию  $a$  по модулю  $n$ , то есть такое число  $x$  ( $0 \leq x < n$ ), что  $a^x \equiv b \pmod{n}$ .

### Формат входных данных

В первой строке входного файла заданы через пробел три целых числа  $a$ ,  $b$  и  $n$  ( $0 \leq a, b, n \leq 10^{12}$ ),  $n \geq 2$ .

### Формат выходных данных

В первой строке выходного файла выведите  $-1$ , если дискретного логарифма не существует. Иначе следует вывести его значение.

Если ответ неоднозначен, разрешается выводить любой.

### Примеры

<code>log.in</code>	<code>log.out</code>
2 4 6	2
1342 134001340034 134	65

## Задача С. Налоги

Имя входного файла: `tax.in`  
Имя выходного файла: `tax.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Король Байтландии следует мировой тенденции и вводит налоги везде, где только может. Недавно он придумал налог на путешествия, который взимается с каждого, кто путешествует по стране.

Каждой Байтландской дороге сопоставлен размер налога за нее. При прохождении через город путешественнику необходимо заплатить налог, равный максимуму из размеров налога за дорогу, по которой он вошел в город, и дорогу, по которой он вышел из города. Также необходимо заплатить за первый и последний город: для них налог равен единственной соответствующей дороге.

Ваш друг Байтазар собирается в путешествие из Байтгорода в Байтополис. Помогите ему спланировать маршрут таким образом, чтобы размер уплаченного налога был минимален.

### Формат входных данных

Первая строка ввода содержит два целых числа  $n$  и  $m$  ( $2 \leq n \leq 100\,000, 1 \leq m \leq 200\,000$ ) — количество городов и количество дорог в Байтландии. Города пронумерованы от 1 до  $n$ .

Следующие  $m$  строк содержат описания дорог.  $i$ -я из этих строк содержит три целых числа  $a_i, b_i, c_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq c_i \leq 1\,000\,000$ ). Это означает, что города  $a_i$  и  $b_i$  соединены двунаправленной дорогой с размером налога, равным  $c_i$  байтлей. Каждая пара соединена не более, чем одной дорогой.

### Формат выходных данных

В первой и единственной строке вывода должно содержаться одно целое число — минимальный размер налога (в байтлях) на путешествие из Байтгорода (т.е. города номер 1) в Байтополис (т.е. город номер  $n$ ). Гарантируется, что существует путь, соединяющий эти два города.

### Примеры

<code>tax.in</code>	<code>tax.out</code>
4 5 1 2 5 1 3 2 2 3 1 2 4 4 3 4 8	12
6 6 6 2 4 4 2 3 3 5 3 2 1 8 4 1 2 4 6 6	13

## Задача D. Полезные ископаемые

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

Ведется проект по освоению планеты соседней звездной системы. Для добычи полезных ископаемых планируется направить на планету несколько партий роботов.

Участок поверхности планеты, на котором планируется добывать полезные ископаемые, представляет собой клетчатый прямоугольник размером  $w$  на  $h$ , клетки участка имеют координаты от  $(1, 1)$  до  $(w, h)$ . В некоторых клетках участка находятся базы специалистов, в которые могут быть доставлены партии роботов. Всего на участке размещено  $s$  баз, и  $i$ -я база находится в клетке с координатами  $(x_i, y_i)$ .

Каждая партия роботов характеризуется тремя параметрами:  $j$ -я партия доставляется на базу  $b_j$ , содержит  $n_j$  роботов и каждый робот партии обладает мобильностью  $m_j$ .

Когда партия роботов доставляется на соответствующую базу, каждый робот этой партии перемещается по поверхности планеты от базы до некоторой клетки. Если мобильность робота равна  $m$ , он может не более  $m$  раз переместиться на одну из восьми соседних клеток.

После того как роботы из всех доставленных партий размещаются на участке, они активируются и начинают добычу полезных ископаемых. В процессе перемещения в одной клетке может одновременно находиться произвольное количество роботов. Однако после активации в каждой клетке должно находиться не более  $q$  роботов.

Руководством проекта получена информация о  $t$  партиях роботов, которые могут быть последовательно отправлены на планету. После доставки всех партий роботов, учитывая их ограниченную мобильность, возможна ситуация, что не удастся разместить роботов на участке так, чтобы в каждой клетке оказалось не больше  $q$  роботов. Поэтому руководство должно выбрать  $k$  первых партий роботов, где  $0 \leq k \leq t$ , которые будут полностью доставлены на соответствующие базы. После этого, если  $k < t$ , следует дополнительно принять  $z$  из  $n_{k+1}$  роботов следующей,  $(k + 1)$ -й партии,  $0 \leq z < n_{k+1}$ .

Все полученные таким образом роботы должны с учетом ограничения на мобильность разместиться на участке таким образом, чтобы в каждой клетке было не более  $q$  роботов. После этого они будут активированы и начнут добычу полезных ископаемых. Разумеется, руководство проекта старается максимизировать количество роботов, которые будут доставлены на планету, поэтому, с учетом описанных ограничений, требуется максимизировать  $k$ , а затем максимизировать  $z$ .

Требуется написать программу, которая по размерам участка, числу  $q$ , описанию расположения баз, а также количеству запланированных партий роботов и их описанию определяет максимальное число  $k$  — количество партий роботов, и затем — максимальное число  $z$  — дополнительное количество роботов из  $(k + 1)$ -й партии, чтобы, доставив роботов на планету, их можно было разместить на участке таким образом, чтобы в каждой клетке оказалось не более  $q$  роботов.

### Формат входных данных

Первая строка входного файла содержит числа  $w, h, s$  и  $q$  ( $1 \leq w, h \leq 10^5, 1 \leq s \leq 4, 1 \leq q \leq 100$ ). Последующие  $s$  строк содержат по два целых числа  $x_i, y_i$  и описывают базы специалистов ( $1 \leq x_i \leq w, 1 \leq y_i \leq h$ ).

Следующая строка содержит число  $t$  — количество партий роботов ( $1 \leq t \leq 100$ ). Последующие  $t$  строк описывают партии роботов и содержат по 3 целых числа:  $b_j, n_j$  и  $m_j$  ( $1 \leq b_j \leq s, 1 \leq n_j \leq w \cdot h \cdot q, 0 \leq m_j < \max(w, h)$ ).

### Формат выходных данных

Требуется вывести два числа:  $k$  и  $z$ ,  $0 \leq k \leq t$ . Если  $k = t$ , то  $z$  должно быть равно 0, иначе должно выполняться условие  $0 \leq z < n_{k+1}$ .

## Пример

стандартный ввод	стандартный вывод
4 3 2 1 1 1 3 2 3 1 4 1 2 9 1 1 12 2	1 7

## Замечание

В приведенном примере описания входных данных следует полностью принять первую партию роботов и дополнительно принять 7 роботов из второй партии.