

## Задача А. Конфеты и преподаватели

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

В ЛКШ начался Новый год, и дети начали дарить друг другу конфеты. В ЛКШ  $n$  детей, среди них есть  $m$  пар детей, которые дружат. Отношение дружбы симметрично: если ребёнок  $i$  дружит с ребёнком  $j$ , то и ребёнок  $j$  дружит с ребёнком  $i$ .

Дети по очереди дарят конфеты всем своим друзьям, при этом на  $i$ -м шаге ребёнок  $v_i$  дарит каждому своему другу по  $k_i$  конфет из неограниченных запасов Деда Мороза. Один и тот же ребёнок может дарить конфеты своим друзьям несколько раз.

Для контроля над процессом время от времени преподаватели могут подойти к какому-то ребёнку и узнать, сколько всего конфет ему подарили на данный момент. Поскольку преподаватели не хотят считать конфеты вручную, они попросили вас помочь им автоматизировать этот процесс.

### Формат входных данных

В первой строке входного файла находятся два целых числа  $n$  и  $m$  ( $1 \leq n$ ;  $0 \leq m$ ;  $n, m \leq 100\,000$ ) — количество детей и количество пар друзей, соответственно.

Все дети пронумерованы числами от 1 до  $n$ . В каждой из следующих  $m$  строк находятся по два числа — пара друзей.

В следующей строке находится число  $q$  ( $1 \leq q \leq 100\,000$ ) — количество действий, которые необходимо обработать. В следующих  $q$  строках находятся либо три числа 0,  $v_i$ ,  $k_i$ , означающих, что человек с номером  $v_i$  дарит всем своим друзьям по  $k_i$  конфет ( $1 \leq k_i \leq 1\,000\,000\,000$ ), либо два числа 1,  $v_i$ , обозначающих, что преподаватель хочет узнать количество конфет, подаренных ребёнку с номером  $v_i$ .

### Формат выходных данных

Для каждого запроса в отдельной строке выведите одно число — количество конфет у соответствующего ребёнка в момент, когда к нему подошёл преподаватель.

---

**Примеры**

стандартный ввод	стандартный вывод
2 1 1 2 8 1 1 0 1 5 0 2 3 1 2 0 1 1 1 1 1 2 0 1 1	0 5 3 6
5 6 1 2 2 3 1 3 1 4 3 4 3 5 11 0 3 5 0 2 4 0 3 7 1 3 1 2 0 1 5 1 1 1 2 1 3 1 4 1 5	4 12 16 17 9 17 12

## Задача В. Праздники

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

В стране АУ есть  $n$  городов и  $n$  дорог между ними. Конечно же, из любого города можно добраться до любого другого по дорогам. Скоро случится праздник — день рождения короля. Король хочет провести празднование своего дня рождения во всех городах, в каждом городе празднование должно длиться один день. Король запланировал, чтобы любой житель мог отпраздновать не только в своем городе, но и каком-нибудь соседнем, поэтому не должно быть двух соседних городов, в которых празднование проходит одновременно.

Сейчас король занят подготовкой к празднику, и у него совсем нет времени на составление расписания, поэтому он просит вас написать программу, которая сама для каждого города определит, когда нужно праздновать в этом городе. Праздник можно отметить в любой день, начиная с дня рождения короля, который является первым днем в году. Разумеется, жители не должны много развлекаться, а должны работать для процветания государства, поэтому необходимо минимизировать номер последнего дня празднования.

### Формат входных данных

В первой строке задано число  $n$  ( $1 \leq n \leq 10^5$ ) — количество городов в стране АУ. В следующих  $n$  строках заданы числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ) — города, между которыми есть дороги. Никакая дорога не соединяет город сам с собой, между любой парой городов не более одной дороги.

### Формат выходных данных

Для каждого города выведите номер дня, в который следует отмечать день рождения короля. Если возможных решений несколько, выведите любое.

### Примеры

стандартный ввод	стандартный вывод
4 1 2 2 3 3 1 2 4	3 2 1 1

## Задача С. Волшебные шары

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

У мага есть  $n$  волшебных шаров, пронумерованных от 1 до  $n$  и расположенных вдоль прямой в порядке номеров. Каждый шар имеет некоторый заряд, который выражается целым числом. В нулевой день заряд каждого шара равен нулю. Каждую полночь заряд каждого шара увеличивается ровно на один. Таким образом, в  $i$ -й день каждый шар имеет заряд  $i$ .

Поскольку магические шары устарели и больше не применяются, их необходимо деактивировать. Про каждый шар известно, при каких значениях заряда его можно безопасно деактивировать:  $i$ -й шар можно деактивировать, если его заряд находится в диапазоне от  $a_i$  до  $b_i$ , включительно.

Для деактивации шаров маг может применять заклинание. Заклинание характеризуется двумя числами  $j$  и  $d$  и применяется к отрезку из одного или нескольких расположенных рядом шаров. Заклинание с параметрами  $j, d$ , где  $1 \leq j \leq n$ ,  $1 \leq d \leq n - j + 1$ , можно применить, если шары с номерами  $j, j + 1, j + 2, \dots, j + d - 1$  можно безопасно деактивировать. В результате применения такого заклинания все эти шары становятся деактивированы. Каждый день маг может применить произвольное число заклинаний. Но при этом каждый шар можно деактивировать всего один раз.

Помогите магу деактивировать все шары, применив суммарно минимальное число заклинаний.

### Формат входных данных

В первой строке входных данных дано  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ ) — число шаров, которое есть у мага. В следующих  $n$  строках дано описание каждого шара. Каждый шар описывается двумя числами  $a_i$  и  $b_i$  ( $1 \leq a_i \leq b_i \leq 10^9$ ).

### Формат выходных данных

В первой строке выходных данных выведите  $z$  — число заклинаний, которое необходимо применить магу.

На следующих  $z$  строках выведите описания заклинаний, каждое в отдельной строке. Каждое заклинание описывается тремя числами: номером первого шара  $j$ , числом шаров  $d$ , к которым оно применяется, и номером дня  $t$ , в который необходимо применить заклинание.

### Примеры

стандартный ввод	стандартный вывод
4	3
1 2	1 2 2
2 2	3 1 3
3 3	4 1 4
4 4	

## Задача D. Запросы

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

На Новый Год Саше подарили массив, состоящий из  $n$  положительных чисел,  $i$ -е число в нем равно  $a_i$ . Миша выбрал  $m$  подотрезков в этом массиве,  $i$ -й из них содержит все элементы с  $L_i$ -го по  $R_i$ -й включительно, длина каждого отрезка не менее двух.

Когда Миша придет, он посчитает максимумы на каждом из этих отрезков, а результаты сложит. Именно столько конфет должна будет отдать ему Саша. Пока Миша не вернулся, Саша может успеть заменить одно число в массиве на 0. Помогите ей минимизировать количество отданных конфет!

### Формат входных данных

В первой строке входных данных содержится число  $n$  ( $2 \leq n \leq 10^5$ ) — количество элементов в массиве. Во второй строке содержатся  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива. В третьей строке содержится число  $m$  ( $0 \leq m \leq 10^5$ ) — количество отрезков запросов. Далее  $m$  строк содержат по два числа  $L_i$  и  $R_i$  ( $1 \leq L_i, R_i \leq n$ ,  $R_i - L_i \geq 1$ ) — концы отрезка запроса.

### Формат выходных данных

В единственной строке выходных данных выведите минимальное количество конфет, которое придется отдать Саше.

### Примеры

стандартный ввод	стандартный вывод
6 1 2 3 4 1 1 4 1 3 3 5 4 6 1 2	9

## Задача Е. Дерево

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

Дано дерево, содержащее  $n$  вершин. Любое ребро может быть красным или синим, изначально все ребра красные. Требуется обработать  $m$  запросов, каждый из которых одного из двух типов.

1. Изменить цвет всех ребер на пути между двумя данными вершинами на противоположный.
2. Определить цвет заданного ребра.

### Формат входных данных

На первой строке входного файла дано целое число  $n$  - размер дерева. На следующих  $n - 1$  строках дано по два числа  $a$  и  $b$  — номера двух вершин, соединенных ребром. Вершины нумеруются целыми числами от 1 до  $n$ . На следующей строке дано целое число  $m$  — количество запросов.

На следующих  $m$  строках даны символ и два числа через пробел:  $type\ u\ v$ . Для запросов первого типа  $type = 'U'$ ,  $u$  и  $v$  — вершины, на пути между которыми изменяются цвета ребер. Для запросов второго типа  $type = '?'$ ,  $u$  и  $v$  — вершины, соединяющие ребро, цвет которого необходимо определить. Гарантируется, что для любого запроса второго типа в исходном графе существует ребро, соединяющее две соответствующие вершины.

### Формат выходных данных

В выходном файле нужно через пробел для каждого запроса второго типа вывести 1, если цвет соответствующего ребра красный, либо 0, если цвет соответствующего ребра синий.

### Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3 4 U 2 3 U 3 1 ? 1 2 ? 3 2	0 1