

## Задача А. Связность графа 2

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

Дан связный неориентированный граф. Вам поступают запросы вида: нужно найти количество компонент связности после удаления некоторого количества ребер.

### Формат входного файла

Первая строка входного файла содержит два числа —  $N$  и  $M$  ( $1 \leq N \leq 10\,000$ ,  $1 \leq M \leq 100\,000$ ), обозначающие число вершин и число ребер, соответственно. Следующие  $M$  строк содержат описания ребер. Каждая строка состоит из двух чисел  $a$  и  $b$  — номера вершин, соединяемых соответствующим ребром. Вершины графа нумеруются с единицы. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Следующая строка содержит единственное число  $K$  ( $1 \leq K \leq 10\,000$ ), обозначающее число запросов. Следующие  $K$  строк содержат описания запросов. Каждое описание начинается с числа  $C$  ( $1 \leq C \leq M$ ), обозначающее число ребер в запросе, далее следуют  $C$  чисел, обозначающих номера ребер, входящих в запрос. Все ребра, входящие в запрос, являются различными. Гарантируется, что количество суммарное число ребер во всех запросах не превышает 200 000.

### Формат выходного файла

Для каждого запроса выведите единственную строку. В  $i$ -ой строке должно содержаться число компонент связности в графе.

### Примеры

<code>disconnected.in</code>	<code>disconnected.out</code>
4 5	1
1 2	2
2 3	1
3 4	
4 1	
2 4	
3	
1 5	
2 2 3	
2 1 2	

## Задача В. Ферма

Имя входного файла: `segtree2d.in`  
 Имя выходного файла: `segtree2d.out`  
 Ограничение по времени: 10 секунды  
 Ограничение по памяти: 256 мегабайт

Настала весна и фермер решил заняться удобрением своего земельного участка размерами  $x \times y$  метров. Для этого он закупил удобрения. До начала посевов остаётся  $n$  дней, и фермер хочет успеть сделать как можно больше.

За день фермер может одну из следующих вещей:

- увеличить продуктивность прямоугольного участка земли со сторонами, параллельными осям координат с углами  $(x_1, y_1)$  и  $(x_2, y_2)$  на значение  $w$
- посчитать суммарную продуктивность участка  $(x_1, y_1) - (x_2, y_2)$

Удобрять фермер любит сам, а вот заниматься скучными расчетами ему не интересно. Помогите ему в этом.

### Формат входного файла

В первой строке входного файла записаны числа  $x$  и  $y$  ( $1 \leq x, y \leq 1000$ ). В следующей строке написано количество оставшихся до начала посевов дней  $n$  ( $1 \leq n \leq 100000$ ). Следующие  $n$  строк описывают действия фермера в соответственный день в следующем формате:

- $1 \ x_1 \ y_1 \ x_2 \ y_2 \ w$  — фермер удобряет участок. ( $1 \leq x_1 \leq x_2 \leq x$ ,  $1 \leq y_1 \leq y_2 \leq y$ ,  $-10000 \leq w \leq 10000$ )
- $2 \ x_1 \ y_1 \ x_2 \ y_2$  — фермер просит посчитать плодородность участка. ( $1 \leq x_1 \leq x_2 \leq x$ ,  $1 \leq y_1 \leq y_2 \leq y$ )

### Формат выходного файла

Для каждого запроса плодородности участка в отдельной строке выведите плодородность этого участка.

### Примеры

segtree2d.in	segtree2d.out
8 8	3
3	
1 2 2 8 8 2	
1 1 1 2 2 1	
2 2 2 2 2	

## Задача С. И снова сумма...

Имя входного файла: `sum.in`  
 Имя выходного файла: `sum.out`  
 Ограничение по времени: 3 секунды  
 Ограничение по памяти: 64 мегабайта

Реализуйте структуру данных, которая поддерживает множество  $S$  целых чисел, с которым разрешается производить следующие операции:

- $add(i)$  — добавить в множество  $S$  число  $i$  (если он там уже есть, то множество не меняется);
- $sum(l, r)$  — вывести сумму всех элементов  $x$  из  $S$ , которые удовлетворяют неравенству  $l \leq x \leq r$ .

### Формат входного файла

Исходно множество  $S$  пусто. Первая строка входного файла содержит  $n$  — количество операций ( $1 \leq n \leq 300\,000$ ). Следующие  $n$  строк содержат операции. Каждая операция имеет вид либо «+  $i$ », либо «?  $l$   $r$ ». Операция «?  $l$   $r$ » задаёт запрос  $sum(l, r)$ .

Если операция «+  $i$ » идёт во входном файле в начале или после другой операции «+», то она задаёт операцию  $add(i)$ . Если же она идёт после запроса «?», и результат этого запроса был  $y$ , то выполняется операция  $add((i + y) \bmod 10^9)$ .

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до  $10^9$ .

### Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

### Примеры

sum.in	sum.out
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	