

Задача Lucky. Счастливые цифры

Имя входного файла: `lucky.in`
 Имя выходного файла: `lucky.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайта

Школьнику Васе нравятся числа, которые заканчиваются счастливыми для него цифрами k . Поэтому каждый раз, когда он видит какое-нибудь натуральное число n , он сразу пытается подобрать такое d ($d \geq 2$), что число n в системе счисления с основанием d заканчивается как можно большим количеством цифр k .

Требуется написать программу, которая по заданным числам n и k найдет такое d , чтобы число n в системе счисления с основанием d заканчивалось как можно большим количеством цифр k .

Формат входного файла

Входной файл содержит два целых десятичных числа n и k ($1 \leq n \leq 10^{11}$, $0 \leq k \leq 9$).

Формат выходного файла

В выходной файл выведите два числа: d — искомое основание системы счисления и l — количество цифр k , которым заканчивается запись числа n в этой системе счисления. Если искомого d несколько, выведите любое из них, не превосходящее 10^{12} (такое всегда существует).

Примеры

<code>lucky.in</code>	<code>lucky.out</code>
49 1	3 2
7 5	3 0

Пояснение

Комментарий к первому тесту: $49_{10} = 1211_3$.

Комментарий к второму тесту: Ни в одной системе счисления 7 не заканчивается на цифру 5.

Задача Rect-Online. Прямоугольники-online

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

Вася нарисовал на плоскости N различных точек с целыми координатами. Теперь он хочет уметь быстро отвечать, а сколько точек лежит внутри заданного прямоугольника со сторонами, параллельными осям координат.

Формат входного файла

В первой строке задается количество точек N ($1 \leq N \leq 50\,000$). Последующие N строк содержат координаты точек x_i y_i . Следующая строка содержит число запросов M ($1 \leq M \leq 50\,000$). Следующие M строк содержат описания запросов, каждый в формате x_1, y_1, x_2, y_2 ($0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$).

Для каждого запроса нужно сделать две вещи:

- Посчитать число точек в соответствующем прямоугольнике (включая границы).
- Все посчитанные точки стереть.

Формат выходного файла

Для каждого из запросов выведите посчитанное количество точек.

Пример

<code>rect_online.in</code>	<code>rect_online.out</code>
9	4
0 0	5
0 1	0
0 2	
1 0	
1 1	
1 2	
2 0	
2 1	
2 2	
3	
0 0 1 1	
0 0 2 2	
0 0 3 3	

Задача Rect-Online-Easy. Прямоугольники-online (простая)

Задача «Прямоугольники-online» с ограничением 100 символов на количество точек.

Задача Sum2. 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 » задает запрос $sum(l, r)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $add(i)$. Если же она идет 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.

Пример

<code>sum.in</code>	<code>sum.out</code>
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

Задача Sum2-Easy. И снова сумма... (простая)

Задача «И снова сумма...» с ограничением 100 на количество операций.