

## Задача Е. Муравьи

Имя входного файла: e.in  
Имя выходного файла: e.out  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

На координатной сетке расположено прямоугольное поле с противоположными углами в точках  $(0,0)$  и  $(W, H)$ . По линиям сетки внутри поля бегают  $K$  муравьев. Каждый муравей может бежать в одном из 4 направлений и со скоростью 1 клетка в секунду.

Так как муравьи очень гордые создания, они никогда не уступают дорогу. Поэтому, если два муравья сталкиваются лоб в лоб, они мгновенно разворачиваются и продолжают идти в обратном направлении. Если два муравья, бегут перпендикулярно не обращают никакого внимания друг на друга. Если же муравей достигает границы поля, то он также разворачивается и бежит обратно.

Вам дано положение и направление движения всех муравьев в начальный момент времени. Определите их положение и направление движения через  $T$  секунд.

### Формат входного файла

Первая строка входного файла содержит 4 целых числа:  $W, H, K, T$  ( $1 \leq W, H, K \leq 100$ ,  $1 \leq T \leq 10^9$ ).

Каждая из следующих  $K$  строк содержит по 3 целых числа:  $X_i, Y_i, D_i$ , где  $(X_i, Y_i)$  — координаты муравья,  $D_i$  — направление его движения ( $0 < X_i < W$ ,  $0 < Y_i < H$ ,  $1 \leq D_i \leq 4$ ).  $D_i = 1$ , если муравей движется в сторону увеличения  $X$ ,  $D_i = 2$  — в сторону увеличения  $Y$ ,  $D_i = 3$  — в сторону уменьшения  $X$ ,  $D_i = 4$  — в сторону уменьшения  $Y$ .

Все числа в строках разделены пробелом. Все муравьи находятся в разных точках.

### Формат выходного файла

Выходной файл должен содержать ровно  $K$  строк — по одной для каждого муравья в том же порядке, в котором информация о них дается во входном файле. Каждая строка должна содержать по 3 целых числа, разделенных пробелом — координаты муравья и направление его движения.

### Примеры

e.in	e.out
4 4 2 3 1 1 1 3 3 4	4 1 3 3 0 2
4 4 2 4 1 1 1 3 3 4	3 1 3 3 1 2
4 4 2 2 1 1 1 3 1 3	1 1 3 3 1 1
4 4 2 2 2 1 1 3 1 3	1 1 3 4 1 3

### Note

Для 60% баллов  $1 \leq T \leq 1000$ .

## Задача F. Обезьяна и яблони

Имя входного файла: `f.in`  
Имя выходного файла: `f.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

Всем известно, что самый вкусный фрукт на земле — это яблоки. Знает это даже обезьяна Крис. Яблоки растут в лесу, в котором есть много яблонь, расположенных вдоль реки и пронумерованных последовательно начиная с 1. Иногда Крис приходит в лес, проходит несколько стоящих подряд яблонь (выбранный им интервал) и считает количество яблонь со спелыми яблоками среди них. Кроме того, бывает так, что к его очередному приходу на нескольких рядом стоящих яблонях уже созрели яблоки.

Для каждого прихода Криса нужно сказать, сколько яблонь с созревшими яблоками находятся в выбранном им интервале. В начале все яблони не созревшие.

### Формат входного файла

В первой строке входного файла дается целое число  $M$  — количество событий ( $1 \leq M \leq 100000$ ). Следующие  $M$  строк содержат описания событий — по три целых числа  $D_i, X_i, Y_i$  ( $1 \leq D_i \leq 2, X_i \leq Y_i$ ). Если первое число  $D_i = 1$ , то описываемое событие — это приход Криса, а если  $D_i = 2$  — созревание яблок на яблонях. Остальные два числа  $X_i$  и  $Y_i$ , описывают интервал для события.

В вычислении границ интервала также участвует число  $C$ , изначально равное 0. Интервал для события — это интервал от  $X_i + C$  до  $Y_i + C$  включительно. Гарантируется, что  $1 \leq X_i + C, Y_i + C \leq 10^9$ . Если событие — это созревание яблок, то значение  $C$  не меняется, а если приход Криса, то в подсчете границ интервала участвует предыдущее значение  $C$ , а в результате  $C$  становится равным количеству яблонь с созревшими яблоками в указанном интервале.

### Формат выходного файла

Для каждого прихода Криса выведите одну строку, содержащую одно число, — ответ к задаче.

### Примеры

<code>f.in</code>	<code>f.out</code>
3 2 5 8 2 7 10 1 1 10	6
4 2 2 3 1 1 3 2 2 3 1 -1 3	2 4
6 2 1 7 2 10 12 1 7 11 2 11 13 1 8 10 1 15 17	3 2 0

### Note

Для 35% баллов  $M \leq 10\,000, 1 \leq X_i + C, Y_i + C \leq 10^6$ .  
Для 60% баллов  $1 \leq X_i + C, Y_i + C \leq 10^7$ .

## Задача G. Горная трасса

Имя входного файла: `g.in`  
Имя выходного файла: `g.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

В окрестностях Алматы оборудован туристический горный круговой маршрут (старт и финиш находятся в одной точке). Будем моделировать этот маршрут  $n$  ступеньками одинаковой ширины, задавая каждую  $i$ -ю ступеньку ее высотой над уровнем моря  $a_i$  в метрах. Соседние ступеньки могут иметь одинаковую высоту. Сложность трассы определяется суммарным значением спусков и подъемов. Или формально,

$$difficulty = |a_1 - a_2| + |a_2 - a_3| + \dots + |a_{n-1} - a_n| + |a_n - a_1|.$$

Оказалось, что первоначально сконструированная трасса слишком сложна для туристов. Для того чтобы ее упростить, мы можем использовать  $k$  блоков. Ширина каждого блока совпадает с шириной ступеньки, а высота равна 1 метру. Любой из блоков можно положить на любую ступеньку или ранее положенный блок или не использовать совсем.

На сколько таким образом можно уменьшить сложность трассы в целом?

### Формат входного файла

В первой строке входной файла находятся числа  $N$  ( $2 \leq N \leq 10^6$ ) и  $k$  ( $1 \leq k \leq 10^9$ ). В следующей строке находятся  $N$  целых неотрицательных чисел — высоты каждой из ступенек.

### Формат выходного файла

Запишите в выходной файл одно число — максимально возможное уменьшение сложности трассы.

### Примеры

<code>g.in</code>	<code>g.out</code>
4 5 4 3 2 1	4
3 2 1 2 1	2
7 1000 4 3 3 2 3 2 1	8

### Note

В первом примере сложность трассы составляла 6 (три спуска на 1 и подъем на 3 с учетом круговой трассы). Положив один блок на предпоследнюю ступеньку и два на последнюю, мы уменьшим сложность на 4. Этот же ответ можно было получить и с использованием всех 5 блоков. Но улучшить ответ нельзя.

Для 30% баллов  $N \leq 100$ ,  $k \leq 1000$ .

Для 60% баллов  $N \leq 100\,000$

## Задача Н. Подстроки

Имя входного файла: `h.in`  
Имя выходного файла: `h.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

[<http://ru.wikipedia.org/wiki/Биоинформатика>] С тех пор как в 1977 году был секвенирован фаг Phi-X174, последовательности ДНК всё большего числа организмов были дешифрованы и сохранены в базах данных. Эти данные используются для определения последовательностей белков и регуляторных участков. Сравнение генов в рамках одного или разных видов может продемонстрировать сходство функций белков или отношения между видами (таким образом могут быть составлены Филогенетические деревья). С возрастанием количества данных уже давно стало невозможным вручную анализировать последовательности. В наши дни для поиска по геномам тысяч организмов, состоящих из миллиардов пар нуклеотидов используются компьютерные программы. Программы могут однозначно сопоставить (выровнять) похожие последовательности ДНК в геномах разных видов; часто такие последовательности несут сходные функции, а различия возникают в результате мелких мутаций, таких как замены отдельных нуклеотидов, вставки нуклеотидов, и их «выпадения» (делеции). Один из вариантов такого выравнивания применяется при самом процессе секвенирования. Так называемая техника «дробного секвенирования» (которая была, например, использована Институтом Генетических Исследований для секвенирования первого бактериального генома, *Haemophilus influenzae*) вместо полной последовательности нуклеотидов даёт последовательности коротких фрагментов ДНК (каждый длиной около 600—800 нуклеотидов). Концы фрагментов накладываются друг на друга и, совмещённые должным образом, дают полный геном. Такой метод быстро даёт результаты секвенирования, но сборка фрагментов может быть довольно сложной задачей для больших геномов. В проекте по расшифровке генома человека сборка заняла несколько месяцев компьютерного времени. Сейчас этот метод применяется для практически всех геномов, и алгоритмы сборки геномов являются одной из острейших проблем биоинформатики на сегодняшний момент.

### Формат входного файла

В  $N$  строках входного файла заданы  $N$  строк одинаковой длины  $L$ . Данные строки содержат только символы английского алфавита [ $A - Z a - z$ ].  $2 \leq N \leq 100\,000$ ,  $2 \leq L \leq 100\,000$ . При этом строчные и прописные буквы считаются различными символами. Для уменьшения затрат на ввод/вывод в тестах к задаче  $N \times L \leq 5 \times 1024 \times 1024$ .

### Формат выходного файла

Вывести строку длины  $L + N - 1$  такую, что все заданные во входном файле строки являются ее подстроками, начинающимися с разных позиций. Гарантируется, что такая строка существует. Если возможны несколько ответов, выведите любой.

### Примеры

<code>h.in</code>	<code>h.out</code>
abb bba	abba

### Note

Для 20% баллов в ответе все буквы разные.