

Задача А. Эксперименты с Горлумом

Имя входного файла: `expgorl.in`
Имя выходного файла: `expgorl.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Совсем недавно Ержан изобрел лазер, способный измерять расстояние до отдаленных объектов. Но, как и любое изобретение, лазер нужно протестировать и, желательно, на живом движущемся объекте (не спрашивайте, какова конечная цель данного лазера). Поскольку использовать мышей старомодно, Ержан отправился в запретный лес на поиски подходящего существа.

Как Ержан в лесу поймал Горлума — уже совсем другая история! Главное, что подопытный для лазера нашелся. Существо это зовут Горлум и, несмотря на то, что после долгих тренировок Горлум может строго выполнять команды, будучи существом раздолбаистым, количество команд ограничено пятью, для удобства обозначенными символами латинского алфавита:

- “L” — Горлум делает шаг влево — переход из точки (x, y) в точку $(x - 1, y)$.
- “R” — Горлум делает шаг вправо — переход из точки (x, y) в точку $(x + 1, y)$.
- “F” — Горлум делает шаг вперед — переход из точки (x, y) в точку $(x, y + 1)$.
- “B” — Горлум делает шаг назад — переход из точки (x, y) в точку $(x, y - 1)$.
- “I” — Горлум достает из своего кармана золотое кольцо со светящимися надписями и никуда не двигается.

Для эксперимента Ержан поставил свой лазер в точке $(Laser_x, Laser_y)$ на плоскости в Евклидовом пространстве. Так же, Ержан выучил Горлума выполнять список команд T , где T — это строка содержащая символы команд в порядке их исполнения. Горлум стартует в точке $(Gorlum_x, Gorlum_y)$.

Ваша задача найти максимальное и минимальное расстояния до Горлума, зафиксированные лазером. Ваш ответ будет засчитан правильным, если абсолютная или относительная погрешности двух чисел не превышают 10^{-9} .

Формат входного файла

В первых двух строках входного файла даны число $K \leq 10^5$ и строка S ($|S| \leq 10^4$), состоящая из символов “LRFBI”. Для получения строки-списка-команд T , K раз сконкатенируйте строку S (иными словами $T = S^K$).

В последних двух строках даны две пары чисел: координаты расположения лазера $(Laser_x, Laser_y)$ и старта Горлума $(Gorlum_x, Gorlum_y)$. Все координаты — целые числа, по модулю не превышающие 10^4 .

Формат выходного файла

Два вещественных числа — минимальное и максимальное зафиксированные расстояния. Абсолютная или относительная погрешности чисел не должны превышать 10^{-9} .

Примеры

<code>expgorl.in</code>	<code>expgorl.out</code>
100000 LRFBI 10000 10000 10000 10000	0.0000000000000 1.0000000000000

Note

В 40% тестов $|S| \leq 2 \times 10^3$, $K \leq 2 \times 10^4$.

Задача В. Кубик Риддика

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

Как известно, самой продаваемой игрушкой в истории является Кубик Рубика. Всего за всю историю было продано более 350 миллионов штук.

Один казахский бизнесмен решил повторить этот успех, создав упрощенную версию этой головоломки. Кубик Риддика представляет из себя прямоугольник размером $N \times M$, состоящий из ячеек 1×1 , каждая из которых раскрашена в какой-то цвет. За один ход разрешается циклически сдвигать любую строку или любой столбец в любую сторону на одну ячейку. Вот например, как сдвигается 2-я строка вправо и 3-й столбец вверх:

1 2 3 4		1 2 3 4		1 2 3 4		1 2 7 4
5 6 7 8	=>	8 5 6 7		5 6 7 8	=>	5 6 11 8
9 10 11 12		9 10 11 12		9 10 11 12		9 10 3 12

Решением головоломки является такая конфигурация, в которой в каждой строке либо в каждом столбце все ячейки одного цвета (выигрышная конфигурация).

Для начала продаж головоломки необходимо сделать оценку ее сложности, чем Вы и должны заняться. Для вычисления сложности будем использовать упрощенные правила: Вы должны сдвинуть сначала какие-то столбцы (возможно никакие), а затем — какие-то строки (возможно никакие).

Вам будет дано описание конфигурации Кубика Риддика. Если из заданной конфигурации возможно по упрощенным правилам получить выигрышную, то сложность текущей конфигурации считается равной минимальному количеству ходов, необходимых для достижения выигрышной. Если же получить выигрышную конфигурацию по этим правилам невозможно, считается, что сложность равна 100500 (головоломка может быть все еще решаемая по обычным правилам, но она слишком сложная).

Формат входного файла

Первая строка входного файла содержит два целых числа N и M ($1 \leq N, M \leq 5$). Следующие N строк содержат по M целых чисел каждая — описание головоломки. Каждое число — это номер цвета, в который раскрашена соответствующая ячейка. Номера цветов — целые числа от 1 до 100. Не гарантируется, что приведенное описание соответствует головоломке, решаемой даже обычными правилами.

Формат выходного файла

Выведите одно целое число — сложность заданной конфигурации головоломки.

Примеры

<code>riddicks.in</code>	<code>riddicks.out</code>
2 3 1 2 1 2 3 3	2
2 3 2 2 1 1 2 1	100500

Задача С. Школы

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

Аким области решил открыть музыкальные и спортивные школы в некоторых городах области. Всего в области N городов. Для каждого города известно, сколько детей, желающих учиться в музыкальной школе, и сколько детей, желающих учиться в спортивной школе, живут в этом городе. Аким хочет, чтобы общее количество музыкальных школ было M , и общее количество спортивных школ было S . Поэтому было решено не открывать в одном городе более одной школы (возможно, в некоторых городах не откроют ни одной школы).

Советнику акима было поручено разработать план постройки школ, чтобы максимизировать общее количество обучающихся детей в этих школах по всей области. Помогите ему в этой нелегкой задаче.

Формат входного файла

В первой строке находятся три целых числа N ($1 \leq N \leq 300000$), M , S ($0 \leq \min(M, S)$ и $M + S \leq N$) — количество городов в области, количество музыкальных школ и спортивных школ которые намерен открыть аким соответственно.

В каждой из следующих N строк по два целых числа A_i ($1 \leq A_i \leq 10^5$) и B_i ($1 \leq B_i \leq 10^5$) — количество детей, желающих учиться в музыкальной школе и спортивной школах в i -м городе соответственно.

Формат выходного файла

Выведите одно целое число — количество детей, обучающихся по всей области, в оптимальном плане.

Примеры

<code>school.in</code>	<code>school.out</code>
3 1 1 5 2 4 1 6 4	9
7 2 3 9 8 10 6 3 5 1 7 5 7 6 3 5 4	38

Задача D. Особый граф

Имя входного файла: `specialg.in`
Имя выходного файла: `specialg.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дан ориентированный граф состоящий из N вершин. Особенность графа заключается в том, что из каждой вершины исходит максимум одно ребро. Ваша задача заключается в том чтобы уметь выполнять два вида запросов:

- 1 a — удалить ребро исходящее из вершины a . Гарантируется что несуществующее ребро не удаляется. $1 \leq a \leq N$
- 2 $a b$ — вывести кратчайшее расстояние из вершины a в вершину b , если таковое существует, иначе вывести “-1” без кавычек. $1 \leq a, b \leq N$

Формат входного файла

В первой строке входного файла задано натуральное число $N \leq 10^5$ — число вершин графа.

В следующей строке задано N целых чисел, i -ое число $next_i$ ($0 \leq next_i \leq N$) — вершина, в которую ведет ребро из вершины i . Если $next_i = 0$, считайте, что из вершины i нет ребра.

В третьей строке задано число $M \leq 10^5$ — количество запросов.

В последующих M строках заданы запросы, соответствующие описанию из условия.

Формат выходного файла

Для каждого запроса вида 2 $a b$ выведите ответ на одной строке, в том порядке, в котором заданы запросы.

Примеры

specialg.in	specialg.out
6	4
3 3 4 5 6 4	2
6	-1
2 1 6	-1
2 1 4	-1
2 1 2	
1 3	
2 1 6	
2 1 4	
2 1 3	
4	1
4 4 1 3	3
5	1
2 2 4	
2 2 1	
1 4	
1 2	
2 3 1	

Note

В 50% тестов $N \leq 2 \times 10^3, M \leq 2 \times 10^4$.

Задача Е. Сумасшедшая старушка

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

Следующая задача является вариацией известной головоломки из теории вероятностей. Есть N мест в самолете и все места проданы пассажирам. Пассажирам предлагается входить по одному, в порядке своих мест, от 1 до N (пассажир с номером места 1 входит в первый, следующий входит пассажир с номером места 2, и так далее ...).

Как вы уже догадались, сумасшедшие старушки ненавидят очереди и предписания, поэтому они обычно идут впереди всех и входят первыми (несмотря на то, что у них может не быть первого места), кроме того, они занимают любое место, которое им нравится (случайно, может получится так, что они выберут свое собственное место).

После того как старушка заняла место, следующие пассажиры входят в порядке их мест (как описано выше) и рассаживаются следующим образом:

- если его/ее место пусто, то он/она занимает свое место.
- но если его/ее место уже занято, он/она занимает любое из оставшихся мест.

Какое место достанется последнему пассажиру? Ответ прост: последний пассажир сядет либо на свое место либо на место сумасшедшей старушки!

Предположим, что сумасшедшая старуха занимает место j , которое не является ее собственным. Когда войдет пассажир, который был назначен на место j , он/она занимает какое то из свободных мест. Эта ситуация может повториться со следующими пассажирами. Но когда какой то пассажир сядет на место сумасшедшей старушки, очевидно (!) все следующие пассажиры сядут на свои собственные места. В этой задаче мы хотели бы узнать изначальное место сумасшедшей старушки.

Формат входного файла

Входной файл может содержать несколько тестов. Первая строка входного файла содержит целое число T — число тестов ($1 \leq T \leq 10$). Каждая из следующих T строк содержит $N + 1$ чисел, разделенных пробелом, сначала N ($1 \leq N \leq 10^3$), далее N чисел: места которые были заняты пассажирами в порядке их входа (p_1, p_2, \dots, p_n - где p_i означает, что пассажир, который зашел i -м занял место p_i). Гарантируется, что входные данные правильные, т.е. удовлетворяют условиям задачи.

Формат выходного файла

Выведите место сумасшедшей старушки, если оно может быть определено однозначно, в противном случае, выведите 0.

Примеры

<code>crazy.in</code>	<code>crazy.out</code>
2	0
2 2 1	1
4 2 3 1 4	

Note

В случае 2 2 1 мы не можем определить ответ однозначно, поскольку пассажир 1 является сумасшедшей старушкой, которая заняла чужое место, и пассажир 2 также может быть сумасшедшая старушка, которая села на свое место.

Задача F. Элитная нора

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

Устав от приключений, хоббит Бильбо решил заняться строительством новой норы. Холм, на котором Бильбо планирует купить себе участок, имеет прямоугольную форму, и его можно представить в виде таблицы из N строк и M столбцов (строки и столбцы нумеруются с единицы, номер 1 имеют самая верхняя строка и самый левый столбец). Каждая клетка таблицы соответствует квадратному участку размером 1 на 1 метр. Хоббиты любят четкие и простые формы, поэтому Бильбо твердо намерен купить себе участок прямоугольной формы, со сторонами параллельными границам холма. Строго говоря он выбирает четыре числа x_1, x_2, y_1, y_2 ($x_1 \leq x_2, y_1 \leq y_2$) и покупает все клетки (x, y) , такие что $(x_1 \leq x \leq x_2)$ и $(y_1 \leq y \leq y_2)$.

Недавние приключения принесли Бильбо столько золота, что цена участков его не очень волнует, но при этом он сильно печется о своей репутации, поэтому он хочет, чтобы цена самой дешевой клетки, которую он купит, была как можно больше. Вместе с тем, нашему хоббиту крайне важен простор, поэтому площадь участка должна быть больше, либо равна K . Если существуют несколько участков с равной минимальной стоимостью клетки, Бильбо предпочтет тот, площадь которого больше.

Помогите Хоббиту выбрать наиболее подходящий для него участок.

Формат входного файла

В первой строке входного файла записаны три числа N, M и K , разделенные ровно одним пробелом — количество строк, количество столбцов и минимальная площадь участка, на которую согласен Бильбо, соответственно. Следующие N строк содержат по M чисел каждая — стоимости клеток соответствующей строки таблицы. Таким образом, число стоящее на месте j в строке $i + 1$, соответствует клетке с индексом (i, j) . N и M натуральные и не превосходят 1000, K натуральное и не превосходит размера таблицы, все стоимости находятся в диапазоне от 1 до 10^9 включительно.

Формат выходного файла

В качестве ответа выведите два числа: сначала максимально возможную минимальную стоимость клетки участка, затем максимально возможную площадь такого участка.

Примеры

burrow.in	burrow.out
3 3 3 1 1 1 1 2 2 1 2 2	2 4
1 10 5 4 3 2 5 10 7 6 5 1 100	5 5
3 5 2 5 7 5 5 5 8 5 5 7 5 8 5 8 8 8	8 3

Note

В 40% тестов $M, N \leq 200$.

В 64% тестов $M, N \leq 400$.

Задача G. Круглые слова

Имя входного файла: `rowords.in`
Имя выходного файла: `rowords.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 128 мегабайт

После недавнего конца света Азамат, наконец, выучил алгоритм нахождения наибольшей общей подпоследовательности (LCS — longest common subsequence), и ему сразу стало интересно: как будет работать алгоритм, если слова будут круглыми?

В круглых словах не имеет значения с какого символа они начинаются и в каком направлении читаются. Например, слово “algorithm” можно прочитать как “rithmalgo” и как “oglamhtir”.

Если для слов “algorithm” и “grammar” в обычном представлении длина наибольшей общей подпоследовательности равна 3 (слово “grm”), то для круглого варианта этих же слов длина наибольшей общей подпоследовательности равна 4 (слово “grma”).

Азамат быстро убедился, что известный алгоритм не может получить правильный ответ для круглых слов. Напишите программу, которая сделает это.

Формат входного файла

В двух строках задается по одному слову. Слова не пустые и длина каждого символа не более 2000 символов.

Формат выходного файла

В единственной строке выведите одно целое число — длину наибольшей общей подпоследовательности заданных круглых слов.

Примеры

<code>rowords.in</code>	<code>rowords.out</code>
algorithm grammar	4

Задача Н. Торговля

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

Вдоль трассы Алматы-Тараз есть N населенных пунктов, пронумерованных числами от 1 до N . В начале зимы M неизвестных торговцев привезли из неизвестного аула вязаные шапки и начали ими торговать в этих населенных пунктах. У этих торговцев есть два принципа: не торговать в одном месте более одного раза (один день) и с каждым днем увеличивать цену на шапку.

Более формально каждый i -й торговец:

1. Начинает торговать в населенном пункте L_i со стартовой ценой на одну шапку X_i
2. Каждый день переходит в соседний населенный пункт, то есть, если вчера он торговал в населенном пункте j , то сегодня торгует в населенном пункте $j + 1$
3. Каждый день увеличивает цену на 1, то есть, если вчера цена на его шапки была x , то сегодня цена $x + 1$
4. Завершает торговать в населенном пункте R_i (при этом в пункте R_i торговля происходит).

Наша задача для каждого населенного пункта определить максимальную цену на одну шапку за всю историю.

Формат входного файла

В первой строке находятся два целых числа N ($1 \leq N \leq 300000$) и M ($1 \leq M \leq 300000$) — количество населенных пунктов и количество торговцев соответственно.

В каждой из следующих M строк находятся по три целых числа L_i , R_i ($1 \leq L_i \leq R_i \leq N$) и X_i ($1 \leq X_i \leq 10^9$) — номера начального и конечного населенных пунктов и начальная цена на шапку для i -го торговца соответственно.

Формат выходного файла

В выходной файл выведите N целых чисел, разделенных пробелом, — где i -ое число равно максимальной цене на одну шапку за всю историю продаж i -ого населенного пункта. Если в каком-то населенном пункте никто не торговал шапками, то для этого населенного пункта выведите 0.

Примеры

trading.in	trading.out
5 2 1 3 2 2 4 6	2 6 7 8 0
6 4 4 4 3 1 2 5 5 6 1 6 6 1	5 6 0 3 1 2