

Задача А. Хоббит, или Туда и обратно

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

Бильбо вернулся домой после необычного путешествия и решил написать об этом книгу с названием «Алая книга Западных пределов». Во время своего путешествия он побывал в пещерах под Мглистым хребтом, в тюрьме эльфов Лихолесья, встретил дракона на Одинокой горе и побывал во многих переделках.

Всего Бильбо побывал в n интересных местах. Все интересные места, а также его дом находятся на одной прямой. Его дом находится на прямой в точке с координатой 0, а i -ое интересное место удалено от дома Бильбо на x_i километров. Между каждой парой интересных мест i и j есть скрытый путь, который не проходит через другие интересные места. Длина этого пути равна $|x_j - x_i|$, а опасность равна $c_{i,j}$. Между i -ым интересным местом и домом также есть скрытый путь, длина которого — x_i , а опасность — 0

Бильбо побывал ровно один раз в каждом интересном месте и затем вернулся домой. Известно, что среди всех возможных путешествий Бильбо выбрал кратчайшее, а среди кратчайших — самое безопасное путешествие. Помогите Бильбо узнать длину путешествия, его опасность и порядок посещения интересных мест.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 1000$) — количество интересных мест. Во второй строке дано n чисел x_i ($1 \leq x_i \leq 10^9$), где x_i — удаленность i -того места от дома Бильбо в километрах. Все x_i различны.

В каждой из следующих n строчек содержится по n чисел. j -ое число в i -ой строчке — $c_{i,j}$, опасность скрытого пути между местами i и j ($0 \leq c_{i,j} \leq 10^6$, $c_{i,j} = c_{j,i}$, $c_{i,i} = 0$).

Формат выходного файла

В первой строке выведите два числа — длину и опасность пути Бильбо. Во второй строке выведите n чисел — номера интересных мест в порядке посещения.

Примеры

<code>hobbit.in</code>	<code>hobbit.out</code>
3	6 2
1 2 3	1 3 2
0 10 1	
10 0 1	
1 1 0	

Задача В. Алхимия

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

Много лет алхимики пытались провести трансмутацию с помощью Философского камня. Алхимик Влачур вплотную подобрался к этому! У Влачура имеется прямоугольная металлическая сетка из $(n + 1) + (m + 1)$ железных стержней размером n на m , где горизонтальные стержни пронумерованы числами от 0 до n снизу вверх, а вертикальные — от 0 до m слева направо.

Для проведения трансмутации Влачур совершает следующие операции: он берет какой-то стержень и полностью окрашивает его волшебной краской. Для того, чтобы понять, насколько он близок к достижению своей цели, ему необходимо знать, у скольких квадратов единичной площади его сетки раскрашено ровно t ($0 \leq t \leq 4$) сторон. Помогите ему в этом!

Формат входного файла

В первой строке входного файла находятся три числа n , m и q ($1 \leq n, m \leq 10^9, 1 \leq q \leq 10^5$) — размеры сетки Влачура и количество выполненных им операций. В следующих q строках находятся описания операций.

Если первый символ в строке — «h», то в той же строке находится число x ($0 \leq x \leq n$), и эта операция означает, что Влачур красит горизонтальный стержень с номером x своей волшебной краской. Аналогично, строчка вида «v» y ($0 \leq y \leq m$) означает, что он красит вертикальный стержень с номером y .

Строчка с единственным символом «?» означает, что вам следует вывести пять чисел: для всех t от 0 до 4 выведите число a_t — сколько подпрямоугольников в сетке имеют ровно t закрашенных сторон.

Формат выходного файла

Для каждой операции вида «?» выведите пять чисел — искомые количества прямоугольников.

Примеры

<code>painting.in</code>	<code>painting.out</code>
<code>3 3 4</code>	<code>0 2 5 2 0</code>
<code>h 1</code>	
<code>v 1</code>	
<code>h 2</code>	
<code>?</code>	

Задача С. Эльфийская пирамидка

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

Когда гном Гимли был маленьким и гостил у эльфов в Лориэне, ему подарили мифриловую пирамидку, состоящую из стержня и нанизанных на него колец. Однако, перед входом в родные Железные Холмы, он столкнулся с проблемой: пирамидка оказалась слишком высокой и не пролезала во входные врата. Сопровождающий их эльф посоветовал ему снять верхнее кольцо и попробовать пройти. Гимли, как настоящий гном, послушал эльфа и решил сделать наоборот — снять самое нижнее кольцо.

Однако оказалось, что это не так просто: Гимли может снять кольцо только тогда, когда внутренний радиус всех колец выше него не меньше, чем внешний радиус снимаемого. Если же какое-то кольцо мешает, то сначала необходимо снять его. Поэтому Гимли попросил вас узнать, сколько всего колец ему придется снять, прежде чем получится снять самое нижнее.

Важно, что кольца на пирамидке могут двигаться только вверх.

Формат входного файла

В первой строке входного файла находится одно целое число n ($1 \leq n \leq 10^5$) — количество колец на пирамидке. В следующих n строках записано по два целых числа s_i и w_i ($1 \leq s_i, w_i \leq 10^5, s_i < w_i$) — внутренний и внешний радиусы i -го кольца пирамидки. Самое верхнее кольцо имеет номер один.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество колец, которые необходимо снять, прежде чем можно будет снять нижнее кольцо. В следующей строке выведите номера этих колец в произвольном порядке.

Примеры

<code>pyramid.in</code>	<code>pyramid.out</code>
4	2
10 20	2 3
3 5	
2 4	
1 3	

Задача D. Держать строй!

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

Как известно, в средние века война не была экстраординарным событием. Напротив, многие государства находились в состоянии постоянной войны с соседями на протяжении десятков лет. Мирные же времена, наоборот, считались редкостью, и далеко не каждый человек успевал застать их за свою жизнь.

С развитием науки ее достижения внедрялись и в военное дело. Так появлялись новые виды вооружения, тактики ведения боя, развивалась полевая медицина. Применялся научный подход и при выборе отряда, который отправлялся на очередное задание.

Перед тем, как военачальник назначал отряд, которому предстоит отправиться в бой, он выстраивал в шеренгу всех солдат, которые находились в его подчинении. Затем определялся солдат, который будет командовать выбранным отрядом, после чего определялся сам отряд.

Согласно новейшим на тот момент исследованиям британских алхимиков, отряд должен был состоять из нечетного количества солдат, стоявших в строю подряд друг за другом. Кроме того, командиром отряда должен был оказаться тот солдат, который стоял в центре выбранного набора. Последнее же ограничение состояло в том, что первая буква в имени k -го слева солдата из выбранного отряда должна была совпадать с первой буквой в имени k -го слева солдата в общем строю. При выполнении всех этих требований британские алхимики гарантировали победу в любом бою, поэтому данные требования строго выполнялись.

Следуя сложившимся традициям и принципу «Количество переходит в качество», после выбора командира военачальники обычно давали ему в подчинение наиболее подходящий под ограничения алхимика отряд. Вам же необходимо по данным первым буквам имен солдат, стоящих в строю, определить для каждого из них максимальный размер отряда, которым он сможет командовать.

Формат входного файла

Первая строка входного файла содержит строку S — первые буквы в именах солдат, стоящих в строю. Длина строки не превышает 500 000. Строка состоит только из строчных символов латинского алфавита.

Формат выходного файла

В первой строке выходного файла выведите $|S|$ чисел, разделенных пробелами — максимальный размер отряда, которым сможет командовать соответствующий солдат.

Примеры

<code>army.in</code>	<code>army.out</code>
abcabc	1 3 5 1 3 0

Задача Е. Рыцари

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

Ария Старк — девочка, которой чужды платья, резиночки и куклы, но не чуждо воинское искусство. Она постоянно занимается со своим учителем боем на мечях и хочет в скором времени быть способной постоять за себя и своих любимых.

Сегодня Ария приехала в гости к своему брату на Стену. Главнокомандующий на Стене поручил ей построить и пересчитать всех своих рыцарей. Арии эта задача показалась простой, и она решила ее усложнить. После построения всех рыцарей по росту она начала менять рыцарей местами, поскольку считала, что в новом построении они будут гораздо эффективнее в борьбе с Одичалыми.

После того, как Ария расположила рыцарей так, как ей хотелось, главнокомандующий захотел определить, а сколько же всего пар рыцарей теперь стоят не по росту. Парой, стоящей не по росту, считается такая пара рыцарей, что рыцарь меньшего роста стоит ближе к началу строя.

Формат входного файла

Первая строка входного файла содержит два числа n и m ($1 \leq n \leq 10^{18}, 1 \leq m \leq 10^5$) — количество рыцарей и количество приказов Арии поменяться местами. Следующие m строк содержат эти приказы, по одному в каждой строке. Приказ состоит из двух различных чисел, разделенных пробелом — позиции двух рыцарей, которые меняются местами. Изначально на первом месте стоит самый высокий рыцарь, а количество пар таких, что меньший рыцарь стоит ближе к началу равно нулю. Рост всех рыцарей различен.

Формат выходного файла

Выведите одно число — ответ на задачу по модулю $10^9 + 7$

Примеры

<code>knight.in</code>	<code>knight.out</code>
5 3 1 3 2 3 2 5	7

Задача F. Где мои драконы?

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

Несмотря на советы Джораха, Дейнерис все-таки решила забрать своих драконов из Дома Бессмертных. В Доме Бессмертных есть круглый коридор, из которого ведут n дверей. Дверь с номером i расположена между дверями с номерами $i-1$ и $i+1$, а дверь номер 1 — между дверями с номерами n и 2; i -тая дверь ведет в комнату номер a_i , и никакие две двери не ведут в одну и ту же комнату.

Зайдя в Дом Бессмертных, Дейнерис встретила Пиата Прейя около первой двери. Он сказал ей, что она может найти своих драконов в комнатах с номерами b_1, \dots, b_k . Чтобы оттянуть время, он хочет, чтобы Дейнерис побывала во всех комнатах. Каждый раз, когда она выходит из очередной комнаты, Пиат Прейя называет ей комнату с номером, которую она еще не посещала, чтобы не вызвать подозрений. А чтобы оттянуть время, он называет комнату, которой нет в списке комнат, в которых ей осталось побывать. Если такой комнаты нет, он ничего не говорит.

Если Дейнерис стоит возле комнаты, которую ей необходимо посетить в данный момент, она заходит туда и ищет там драконов ровно одну минуту. Иначе она идет к соседней двери с большим номером. То есть, от двери с номером i она идет к двери с номером $i+1$, но если она стояла у двери с номером n , то она пойдет к двери с номером 1. Чтобы дойти до соседней двери, Дейнерис также тратит одну минуту.

Дейнерис посетила все n комнат, однако, выйдя из последней, она так и не нашла своих драконов. Поняв, что ее обманывают, она убежала от мага. Через час после этого она нашла своих драконов и освободила их. Тут же вырвавшись на свободу, драконы полностью сожгли Дом Бессмертных.

Ваша задача определить, сколько времени в минутах прошло с момента, когда Дейнерис вошла Дом Бессмертных, до момента, когда дом был полностью уничтожен драконами.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \leq n \leq 10^5$) — количество дверей в коридоре. В второй строке дано n чисел a_i ($1 \leq a_i \leq n$), где a_i — номер комнаты, в которую ведет i -ая дверь.

В третьей строке дано k ($1 \leq k \leq n$) — количество данных Дейнерис в самом начале номеров комнат. В четвертой строке даны k чисел b_i ($1 \leq b_i \leq n$, $b_i \neq b_j$ если $i \neq j$) — номера этих комнат. В пятой строке даны $n - k$ чисел c_i ($1 \leq c_i \leq n$) — порядок номеров остальных комнат, в котором Пиат Прейя сообщает их Дейнерис.

Формат выходного файла

В первой строке выведете искомое время. Во второй строке выведете через пробел n чисел d_i — номера посещенных Дейнерис комнат в порядке посещения.

Примеры

<code>dragons.in</code>	<code>dragons.out</code>
4	67
3 1 4 2	3 1 4 2
2	
2 3	
1 4	

Задача G. Эльфы против орков

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

В связи с началом войны против орков эльфы начали укреплять оборону, чтобы быть готовыми к нападению. В первую очередь они установили ловушки на подступах к своей столице. Орки, в свою очередь, уже собрали отряд и начали наступление.

Издrevле для удобства ведения боя была введена декартова система координат на полях сражений. Обе воюющие стороны используют её — эльфийские ловушки размещаются только в точках с целочисленными координатами, и каждый орк из отряда может находиться только в точке с целочисленными координатами.

Эльфы достаточно хитры и расположили свои ловушки внутри и на границах прямоугольника, повернутого на 45 градусов по часовой стрелке. Орки же, напротив, достаточно прямолинейны и выстроили свой отряд внутри и на границах прямоугольника со сторонами, параллельными координатным осям. В каждой точке с целочисленными координатами прямоугольника эльфов стоит ловушка, а в каждой точке прямоугольника орков стоит один орк. Главнокомандующий эльфов знает расположение прямоугольников на данный момент и он задался вопросом: стоит ли хотя бы один орк на одной из ловушек?

Формат входного файла

В первой строке задано количество тестов t ($1 \leq t \leq 10\,000$). Далее, заданы эти тесты. Каждый тест задан в двух строках. В первой строке заданы координаты левого нижнего и правого верхнего углов прямоугольника орков. Во второй строке заданы координаты нижнего и верхнего углов прямоугольника эльфов.

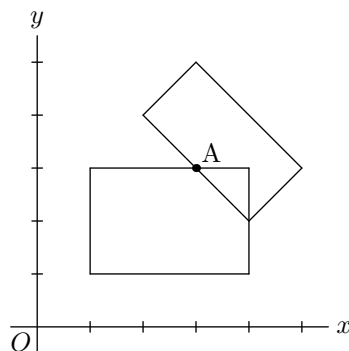
Все точки задаются своими координатами x и y , где x и y — целые числа ($-10^9 \leq x, y \leq 10^9$). Гарантируется, что стороны прямоугольника орков параллельны осям координат, а стороны прямоугольника эльфов повернуты относительно координатных осей на 45 градусов по часовой стрелке. Прямоугольники не могут вырождаться в отрезки и точки.

Формат выходного файла

Если существует точка с целочисленными координатами, в которой расположена ловушка и стоит орк, выведите YES, иначе выведите NO.

Примеры

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



В точке A стоит один из орков, попавших в ловушку в тесте из примера.

Задача Н. Круглый стол

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

Чтобы избежать многочисленных войн, в средние века любили проводить большие собрания, где все могли мирно обсудить интересующие вопросы. Во время такого собрания всех присутствующих рассаживали за большой круглый стол. Скоро пройдет очередное собрание, куда придут представители n стран. Каждая страна отправила на это собрание ровно двух человек — экономиста и политика. Каждый из делегатов хорошо разбирается только в своей сфере, поэтому обсуждать он будет только ее.

Перед организаторами этого собрания возникла проблема рассадить представителей всех стран. Понятно, что нельзя допустить ситуации, чтобы на трех подряд местах сидели люди, которые обсуждают одно и то же (экономику или политику). В этом случае люди начнут формировать коалиции по три человека, и цель собрания не будет достигнута.

Также известно, что экономистам нравится сидеть на красных креслах, политикам — на синих, а на фиолетовых — и те, и другие. Про каждое из кресел, которые стоят вокруг стола, известен его цвет. Главному мудрецу организующей страны было поручено посчитать количество возможных способов рассадки гостей.

Пронумеруем все места вокруг стола числами от одного до $2 \times n$. Два способа считаются различными, если существует такое место i , что в этих способах на нем сидят люди из различных стран или люди, обсуждающие разные стороны политики.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($2 \leq n \leq 1000$) — количество стран-участниц собрания. Во второй строке содержится $2 \times n$ чисел. i -ое число равно 0, 1 или 2, если i -ое кресло фиолетовое, красное или синее соответственно.

Формат выходного файла

В единственной строке выведите количество возможных вариантов размещения представителей стран по модулю $10^9 + 7$.

Примеры

<code>roundtable.in</code>	<code>roundtable.out</code>
2 1 2 1 2	4
3 1 2 0 0 0 2	108

Задача I. «Великая шестерка»

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

В королевстве наступила новая эра. Уже давно ничего не менялось в армии королевства, поэтому король решил выбрать новых предводителей армии.

В подчинении короля есть n воинов. Так сложилось, что у каждого из воинов есть ровно три друга, которые его хорошо понимают и поддерживают.

Король решил из всех своих подчиненных выбрать шесть воинов, которые образуют «Великую шестерку». Воины a_1, a_2, \dots, a_6 образуют великую шестерку если:

- Можно выбрать три воина a_i, a_j и a_k ($i \neq j, j \neq k, i \neq k$), среди которых любые два дружат между собой. Мы назовем их «Главная тройка»
- Каждому из трех воинов «Главной тройки» можно выбрать помощника, причем у каждого из них свой личный помощник
- Помощник воина a_i — это такой воин a_t , который входит в «Главную шестерку», но не входит в «Главную тройку» и является другом a_i

Король задумался, сколькими способами он может выбрать «Великую шестерку». Помогите ему сделать это.

Два способа считаются различными, если отличаются наборы людей в «Великой шестерке».

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 10^5$) — количество воинов в королевстве.

В следующих n строках заданы друзья каждого из воинов. В $i+1$ -й строке три различных числа: номера воинов, которые являются друзьями воина с номером i .

Гарантируется, что данные не противоречивы, и что среди друзей воина нет его самого.

Формат выходного файла

В выходной файл выведите одно целое число: количество способов выбрать «Великую шестерку».

Примеры

<code>sixheroes.in</code>	<code>sixheroes.out</code>
6	1
2 3 4	
1 3 6	
1 2 5	
1 5 6	
3 4 6	
2 4 5	

Задача J. Наследство

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

Среди городов Средиземноморья особенно известен город Сомелот. В городе регулярно проводятся пиры, рыцарские турниры и балы. А для защиты от непрошенных гостей в городе было построено n смотровых башен, с которых зоркие воины следят за окрестностями. Долгое время городом правил мудрый лорд Эревард. Во времена его правления город процветал, а жители города любили и уважали своего правителя. Но время не пощадило и его, и на девяносто третьем году жизни он скончался.

По завещанию, управление городом перешло к его сыновьям: Бенедикту и Теобальту. Но политические взгляды братьев не совпали, они крепко разругались и решили разделить город. Братья решили построить два забора, которые будут отделять новые города друг от друга. Заборы должны состоять из секций, каждая из которых соединяет две смотровых башни. Также заборы должны быть замкнутыми, и каждая башня должна стоять на конце ровно двух секций забора. Ни одна башня не должна остаться вне или внутри новых городов.

Кроме этого, чтобы как можно реже встречаться друг с другом, братья хотят, чтобы у заборов не было общих точек как на границах, так и внутри самих заборов. А также, чтобы в городе с каждой смотровой башни было видно все точки, находящиеся внутри соответствующего города. Считается, что точку видно с башни, если на отрезке между башней и этой точкой нет заборов и других башен.

Для выполнения этой работы были наняты лучшие архитекторы, но они не смогли решить одну простую задачу: как разделить город так, чтобы никому из братьев не было обидно. Архитекторы обратились с этим вопросом к вам. Вам необходимо построить забор так, чтобы разница площадей новых городов была минимальна.

Формат входного файла

Первая строка входного файла содержит число n ($6 \leq n \leq 100$) — число смотровых башен в городе Сомелот. Следующие n строк содержат по два числа x_i, y_i ($1 \leq x_i, y_i \leq 10^9$) — координаты i -ой смотровой башни. Гарантируется, что никакие три башни не лежат на одной прямой и никакие две башни не лежат в одной точке.

Формат выходного файла

В первой и единственной строке выходного файла выведите минимально возможную разность площадей новых городов, с точностью не менее 6 знаков после запятой. Если разделить город так, чтобы были выполнены требования братьев невозможно, выведите `-1`.

Примеры

<code>inheritance.in</code>	<code>inheritance.out</code>
6 0 0 0 1 1 0 3 2 2 2 3 1	0