

Задача А. Заправки

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

Ударим автопробегом по
бездорожью и разгильдяйству!

О. Бендер

Организаторы автопробега Москва–Самара ответственно подошли к организации заправок на маршруте. Они составили карту всех возможных маршрутов и выяснили, что из каждого города, кроме Самары, участники могут поехать по двум другим дорогам в город(а) с большими номерами.

Все участники заинтересованы в победе, поэтому движутся только к цели: они не могут поехать назад той дорогой, по которой приехали, так как знают, что, если будут ехать вперёд, то всё равно доберутся до цели. Также, в каждый город, кроме Самары, из Москвы можно попасть единственным маршрутом.

Организаторы хотят расставить в городах заправки таким образом, чтобы:

1. В Москве и в Самаре была заправка.
2. Если в городе a нет заправки, и из него можно поехать в город b , то в городе b должна быть заправка (автомобили могут и не проехать два перегона между городами на одной заправке).
3. На любом пути из Москвы в Самару было встречено одно и то же число заправок (чтобы все участники были в равных положениях).

Помогите организаторам выяснить, смогут ли они выбрать города, в которых будут организованы заправки, так, чтобы все эти три пункта удовлетворялись.

Формат входного файла

В первой строке задано число n ($2 \leq n \leq 200\,000$) — число городов. Москва имеет номер 1, Самара — n . Далее, в $n - 1$ -ой строках заданы пары чисел a_i, b_i ($i < a_i, b_i \leq n$) — номера городов, в которые можно поехать из i -го.

Формат выходного файла

Выведите «YES», если можно выбрать города с заправками, отвечающие описанным ограничениям, или «NO» в противном случае.

Пример

<code>redblack.in</code>	<code>redblack.out</code>
4 2 3 4 4 4 4	YES
5 2 5 3 4 5 5 5 5	NO

Задача В. Необычные шахматы

Имя входного файла:	<code>chess.in</code>
Имя выходного файла:	<code>chess.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

После провала шахматного турнира в клубе «Картонажник», Остап решил научиться играть в шахматы, чтобы больше такого не случилось. Но игра в обычные шахматы его не привлекает — научиться хорошо играть в них сложно и долго. Поэтому он придумал свои шахматы, в которые еще никто не умеет играть. Он назвал их BenderChess.

Правила игры в BenderChess почти такие же, как и правила игры в обычные шахматы, но, всё-таки, немного отличаются. В шахматах Остапа фигуры ходят так же, как и в обычных шахматах, но зато каждый игрок делает не один ход, а сразу два подряд. «Двойной ход» является корректным, если ни после первого хода, ни после второго хода игроку, делающему ход, не будет объявлен шах. Также нельзя ходить в клетку, занятую фигурой того же цвета. Недопустимо, чтобы в какой-то момент два короля стояли в соседних клетках (соседними называются клетки, у которых есть хотя бы одна общая точка).

Королю объявляется шах, если есть такая фигура, которая будет бить короля после убирания с доски всех фигур кроме неё и короля. Тем не менее, как и в обычных шахматах, нельзя перепрыгивать через другие фигуры во время хода.

Остап решил опробовать свою игру, поэтому стал играть в нее с Кисой. В последней партии игроки перешли в эндшпиль¹: у каждого игрока остались только король, слон и ладья. Остап плохо играет эндшпили, поэтому ему надо, чтобы кто-нибудь закончил за него игру. Но это сложная задача, поэтому он просит Вас хотя бы сделать за него один двойной ход. Помогите ему.

Формат входного файла

В первой строке входного файла даны два числа n, m ($1 \leq n, m \leq 10^3$) — размеры доски. Во второй строке входного файла дано описание белых фигур. Сначала записаны числа x, y ($1 \leq x \leq n, 1 \leq y \leq m$) — координаты короля. Затем в таком же формате записаны координаты ладьи и слона, соответственно.

В третьей строке входного файла дано описание черных фигур в таком же формате.

Гарантируется, что изначально никому не объявлен шах и короли не находятся в соседних клетках. Также, никакие две фигуры не стоят в одной клетке.

Остап играет за белых.

Формат выходного файла

В выходной файл выведите две строки — два хода. Каждый ход выводите в формате x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 \leq n, 1 \leq y_1, y_2 \leq m$), где x_1, y_1 — координаты фигуры перед ходом, x_2, y_2 — координаты фигуры после хода.

Если решения не существует, выведите -1.

Если за один ход можно объявить шах, то на втором ходе можно съесть короля противника.

¹заключительная часть шахматной партии

Пример

chess.in	chess.out
3 3 1 1 2 1 3 1 3 3 2 3 1 3	1 1 1 2 1 2 1 1
3 3 1 1 1 2 2 1 3 3 3 2 2 3	1 2 2 2 2 2 1 2
4 2 3 1 4 1 4 2 1 2 2 2 2 1	-1

Комментарий

Великий комбинатор, естественно, играет за белых.

Задача С. Путешествие

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

После того, как Остап Бендер в очередной раз упустил Александра Ивановича Корейко, он решил взять ситуацию в свои руки. Расставшись со своей командой и удобной во всех отношениях Антилопой Гну, Остап решил подойти к проблеме с умом.

Господин Корейко, дабы не показывать своего состояния, решил убежать от Остапа, передвигаясь по стране исключительно на поездах. Однако, поскольку железнодорожная сеть в нашей стране чрезвычайно запутана, для того, чтобы добраться до конечного пункта назначения, иногда приходится делать пересадки. Известно, что в железнодорожную сеть входит n станций. Между некоторыми станциями есть двусторонние поезда. У каждого поезда есть своя цена.

После длительных размышлений, Остап понял, что, для конспирации, его подопечный путешествовал по стране по следующему правилу: сначала он передвигался по маршрутам, цена которых не превышала A . Затем, когда все стали думать, что господин Корейко совершенно не богат, он решил путешествовать с комфортом. Александр Иванович решил передвигаться по стране маршрутами ценой не менее B . Но все подпольные миллионеры ужасно скупы. Известно, что он затратил минимальное количество средств. Так же известно, что Корейко выехал со станции с номером s и приехал на станцию с номером t .

Остап хочет узнать, насколько уменьшилось состояние Александра Ивановича в результате этого путешествия.

Формат входного файла

В первой строке входного файла даны числа n и m ($1 \leq n, m \leq 10^5$) — количество станций и поездов. Во второй строке даны два числа s, t ($1 \leq s, t \leq 10^5$). В следующих m строках дано описание маршрутов: a, b, c ($1 \leq a, b \leq n, 1 \leq c \leq 10^9$) — номера станций, соединенных маршрутом, и стоимость маршрута. В последней строке даны два числа A, B ($1 \leq A, B \leq 10^9$).

Формат выходного файла

В единственной строке выходного файла выведите ответ на задачу. Если такого пути не существует, выведите -1.

Пример

travel.in	travel.out
4 6 1 4 1 2 4 2 4 4 4 1 6 1 3 7 4 3 10 3 2 2 5 6	6
6 9 2 4 2 6 4 6 4 5 4 5 1 5 3 10 3 2 9 2 5 5 1 5 2 1 6 2 1 2 1 5 3	4

Задача D. Инверсии

Имя входного файла: `inversions.in`
Имя выходного файла: `inversions.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайта

Остапа уже не удивишь бриллиантами в стульях. Вот и сейчас он не удивлен: перед ним стоят n стульев. Остап знает, что в i стуле спрятано a_i бриллиантов, причем все a_i различны и лежат в отрезке от 1 до n .

Но Остапа интересуют не сами бриллианты, а нечто совершенно иное. Он заинтересован в количестве инверсий. Инверсией называется такая пара (i, j) , что $i < j$ и $a_i > a_j$. Считать количество инверсий среди всех стульев Остап научился легко. Теперь перед ним стоит более сложная задача: научиться быстро и без заминки считать количество инверсий среди некоторого количества подряд стоящих стульев. Остап справился, а сможете ли Вы?

Формат входного файла

В первой строке находится целое число n ($1 \leq n \leq 30\,000$) — количество стульев. Во второй строке находятся n целых чисел a_i ($1 \leq a_i \leq n$), разделенных пробелами. Гарантируется, что все эти числа различны.

В следующей строке находится целое число q ($1 \leq q \leq 100\,000$) — число запросов. Следующие q строк содержат по два числа x_i и y_i ($1 \leq x_i, y_i \leq n$), необходимые для генерации границ i запроса. Сами границы определяются как $(x_i + Ans_{i-1} - 1) \bmod n + 1$ и $(y_i + Ans_{i-1} - 1) \bmod n + 1$, где Ans_{i-1} — ответ на предыдущий запрос, либо 0, если i равно 1. Минимальное из этих двух чисел будет левой границей отрезка, а максимальное — правой. $a \bmod b$ равно остатку от деления a на b .

Формат выходного файла

Для запроса i выведите в строке с номером i единственное число: количество инверсий на i -м отрезке.

Примеры

<code>inversions.in</code>	<code>inversions.out</code>
8	4
1 4 6 2 3 8 7 5	6
6	2
1 5	5
7 4	3
4 6	8
8 5	
1 3	
6 5	

Комментарий

Истинные границы запросов выглядят следующим образом:

- 1 5
- 3 8
- 2 4
- 2 7
- 6 8
- 1 8

Задача Е. Дерево

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

Остап Бендер нашел новое применение своим безграничным талантам, а именно, решил удаться в бизнес. На современном рынке полно всяких товаров, однако его это не смущает, ведь он торгует не чем-нибудь, а корневыми деревьями.

Всем известно, что цена корневого дерева — это сумма глубин его листов. Корень дерева имеет глубину 0, а глубина любой другой вершины равна глубине ее предка плюс один. У Остапа никогда не возникало проблем с тем, чтобы определить цену дерева, глядя на него, но вот строить дорогие деревья он не умеет.

У нашего героя есть N вершин, и целых $N - 1$ ребро. Он может построить из них одно, или несколько деревьев, а потом продать. Помогите Остапу, найдите максимальную суммарную стоимость построенных деревьев.

Формат входного файла

Первая строка входного файла содержит единственное число N ($1 \leq N \leq 8\,589\,934\,591$) — количество вершин, которые есть у Остапа.

Формат выходного файла

Выведите одно число — максимальная суммарная цена построенных деревьев.

Примеры

<code>tree.in</code>	<code>tree.out</code>
3	2

Комментарий

В этом примере можно обойтись одним деревом. Пусть корнем дерева будет вершина 1, тогда выгодно провести ребра $1 \rightarrow 2$ и $1 \rightarrow 3$. Стоимость дерева — сумма глубин второй и третьей вершины — $1 + 1 = 2$.

Листом дерева называется вершина, соединенная только со своим предком.

Задача F. Стулья

Имя входного файла: `towers2.in`
Имя выходного файла: `towers2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Однажды Киса Воробьянинов все таки нашел драгоценности тещи, которые были спрятаны в стульях. Он выставил стулья в два ряда и решил нанять людей, которые бы следили за тем, чтобы их никто не украл. Киса, по своей природе, был очень жадным и, хоть он и нашел драгоценностей на миллионы рублей, все равно хотел сэкономить каждую копейку. Поэтому он хотел, чтобы каждый нанятый им человек следил ровно за двумя стульями — одним из первого ряда и одним из второго ряда. И весь процесс охраны должен был выглядеть следующим образом. Каждый нанятый человек сидит какое-то время на одном стуле, а потом быстро бежит к второму стулу и сидит на нем еще какое-то время. Потом он опять бежит к первому стулу и все начинается с начала.

Чтобы два нанятых человека случайно не столкнулись друг с другом, пока бегут от одного стула к другому, Киса понял, что не должно существовать двух человек, для которых бы выполнялось следующее условие. Пусть первый охраняет стул i из первого ряда и стул j из второго, а второй — стул k из первого и l из второго. Тогда условие можно записать так: $i < k$ и $j > l$. Все же остальное он считал нормальным. Например, нормально, что один и тот же стул могут охранять несколько человек. Так же его не волнует общее количество людей, которым он должен будет заплатить за охрану (главное, чтобы каждый из них трудился не покладая рук — тогда совесть Ипполита Матвеевича будет чиста).

Как только Остап Бендер узнал про все это, он понял, что командует парадом сейчас, к сожалению, не он, и нужно срочно взять все в свои руки. В частности, он захотел взять в свои руки два стула с драгоценностями. В левую руку стул из первого ряда, а в правую из второго. Взять только один стул ему не позволит совесть. Но взять он может только такую пару стульев, что не существует человека, который охраняет их одновременно. Конечно, он хочет взять два стула, суммарное количество драгоценностей в которых как можно больше. Будем считать, что великий комбинатор знает, сколько драгоценностей находится в каждом стуле.

Вам предлагается узнать, сколько денег выручит Остап после того, как продаст украденные бриллианты, если известно, что Киса постарался на славу и охрана была выставлена наилучшим способом.

Формат входного файла

В первой строке задано два целых числа n, m ($1 \leq n, m \leq 200\,000$) — количество стульев в первом и втором ряду соответственно. В следующей строке содержится n целых чисел ($1 \leq a_i \leq 10^9$) — стоимости бриллиантов в стульях первого ряда. В третьей строке в аналогичном формате задана информация о стульях второго ряда.

Формат выходного файла

В первой и единственной строке выведите стоимость бриллиантов, которые украл Остап. Если он не смог ничего украсть, выведите 0.

Примеры

<code>towers2.in</code>	<code>towers2.out</code>
2 3 1 2 3 4 5	6

Задача G. Рекламный щит

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

Остап Бендер, великий комбинатор, решил податься в рекламный бизнес. Теперь он делает рекламные щиты. И Остап понимает, что куда дешевле взять старый щит, вырезать из него ненужные куски и получить новый.

Рекламный щит представляет собой табличку с лозунгом. Остап придумал новый лозунг, который можно получить из старого, путем выпиливания нескольких кусков из старого и склеивания этих кусков в том же порядке. К сожалению, места склейки нелицеприятно выглядят, поэтому Остап хочет уменьшить количество кусков, из которых собирается новый щит. Помогите Остапу подсчитать минимальное количество кусков, необходимое для получения нового щита.

Формат входного файла

В первой строке входного файла дана строка s , длины n ($1 \leq n \leq 10^5$) — лозунг на первом щите, состоящий из маленьких английских букв. Во второй строке входного файла дана строка t , длины m ($1 \leq m \leq 10^5$) — лозунг, который хочет получить Остап.

Формат выходного файла

В единственной строке выходного файла выведите минимальное количество кусков, из которых собирается новый щите. Гарантируется, что ответ не превышает 10.

Пример

<code>advertisement.in</code>	<code>advertisement.out</code>
<code>buyourchairs yourhairs</code>	2
<code>goldencow old</code>	1

Задача Н. Билеты в Провал

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

Как-то раз, в Пятигорске, Остап Бендер решил заработать денег. Для этого он начал продавать билеты в открытый для всех Провал. Провал — это глубокий природный колодец-пещера с подземным озером. Всего для распространения у Остапа было 2^n билетов.

У Остапа в роду были программисты, и поэтому все билеты он пронумеровал целыми числами от 0 до $2^n - 1$. После этого, на каждом билете оказалось по три числа — штрих-код, номер серии билета и номер, написанный ручкой Остапа Бендера.

Первым покупателем Остапа оказался программист, который захотел купить два билета, отвечающих следующим условиям:

- побитовое логическое «И» номеров билетов, написанных Остапом, равно нулю
- сумма штрих-кода первого билета и номера серии второго билета максимальна

Поскольку Остап не очень силен в быстрых подсчетах, помогите ему найти два таких билета.

Формат входного файла

В первой строке дано одно натуральное число n ($1 \leq n \leq 20$). Далее в 2^n строках содержится информация о билетах. В i -ой строке входного файла даны два числа x и y ($0 \leq x, y \leq 10^9$) — штрих-код и номер серии билета, номер которого в нумерации Остапа равен $i - 2$.

Поскольку билеты фальшивые, то у разных билетов могут совпадать номер серии и/или штрих-код.

Формат выходного файла

Выведите два числа — номера двух билетов, которые Остап должен продать покупателю. Если возможных ответов несколько, то выведите любой.

Пример

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

Комментарий

Обратите внимание, что ответы 2 1 и 1 2 различны, поскольку штрих-код берется с первого проданного билета, а серия — со второго.

Задача I. Геркулес

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Остап, не жалея сил, гонится по коридорам «Геркулеса» за полуответственным и полупризрачным товарищем Скумбриевичем. Он знает, что весь «Геркулес» состоит из нескольких комнат, которые соединены двусторонними коридорами так, что из любой комнаты можно попасть в любую ровно одним способом (возможно, пройдя при этом через несколько других комнат). В одной из этих комнат находится Скумбриевич, и он не перемещается.

Каждый раз, попадая в новую комнату, он не может вспомнить, из какого коридора он в нее попал. Однако, он видит, что все коридоры, соединяющие эту комнату с соседними, пронумерованы, начиная с единицы, и заканчивая числом коридоров, имеющих в этой комнате. Остап знает, что нумерация коридоров в комнатах со временем не меняется. Весь процесс погони Остапа за Скумбриевичем выглядит следующим образом.

- Остап выбирает, по коридору с каким номером он пойдет в другую комнату
- он переходит в другую комнату по выбранному коридору
- он оказывается в некоторой другой комнате, про которую он снова знает только количество коридоров, имеющих в этой комнате
- внутренний голос Остапа сообщает ему, приблизился ли он к Скумбриевичу или отдалился
- возможно, он наконец-то встречает Скумбриевича, и тогда погоня заканчивается

Помогите Остапу догнать неуловимого Скумбриевича.

Протокол взаимодействия с программой жюри:

Во время взаимодействия вашей программы с программой жюри несколько раз повторяются следующие действия.

- программа жюри сообщает вашей программе количество коридоров k , имеющих в комнате, в которой в данный момент находится Остап
- ваша программа сообщает программе жюри номер коридора, по которому Остап пойдет в другую комнату — целое число от одного до k
- программа жюри сообщает вашей программе:
 - «Warmer», если этим ходом Остап приблизился к Скумбриевичу
 - «Colder», если этим ходом Остап удалился от Скумбриевича
 - «Success», если этим ходом Остап попал к Скумбриевичу
- в случае, если Остап попал к Скумбриевичу, вам необходимо завершить работу своей программы
- в противном случае, описанные действия начинают повторяться сначала

Гарантируется, что количество комнат в «Геркулесе» не превышает 100. Остап может сделать не более 500 000 переходов.

Пример

stdin	stdout
2	1
Colder	1
1	2
Warmer	
2	
Success	

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

Задача J. Отец Федор

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Отец Федор после длительного пребывания на скалах осознал все свои прошлые ошибки и решил вернуться к прежней жизни священника церкви Фрола и Лавра. К сожалению, скалы очень сильно повредили рассудок отца Федора и теперь ему чудятся видения.

Всего в долине расположены n скал, некоторые из которых соединены двусторонними путями. Длина каждого пути является целым неотрицательным числом, не превышающих 10^{15} . На одной из этих скал находится сам отец Федор. На другой скале находится выход, в который Федор хочет попасть. Отец Федор может со своей скалы добраться до любой другой.

Однако, видение сообщило Федору, что сделать это он сможет только в том случае, когда длина кратчайшего пути между скалой, на которой находится он, и скалой, на которой находится выход, будет равна некоторому числу c , удовлетворяющему неравенству $l \leq c \leq r$. Для того, чтобы положение отца Федора было не слишком безысходным, видение дало ему некоторые сверхспособности.

Некоторые из вершин видение назвало особенными. Особенность этих вершин заключается в том, что отец может своим указанием добавить путь произвольного веса между любой парой особенных вершин. Более того, отец Федор может также и удалить любой из путей, которые он когда-то добавил.

Отец Федор находится не в лучшем физическом состоянии, поэтому видение сообщает ему кратчайший путь между его вершиной и выходом каждый раз после того, как он совершает какое-то действие с путями. Помогите отцу Федору выбраться со скал, используя его сверхъестественные возможности.

Протокол взаимодействия с программой жюри:

В первой строке заданы числа n, s, t ($2 \leq n \leq 100, 1 \leq s, t \leq n, s \neq t$) — количество скал, номер скалы, где сидит отец Федор и номер скалы, на которой находится выход.

Во второй строке находятся числа l и r ($0 \leq l \leq r \leq 10^{15}$) — интервал длины кратчайшего пути.

В третьей строке находится число m ($2 \leq m \leq n$) — число особенных скал.

В следующей строке заданы m чисел a_i ($1 \leq a_i \leq n$) — номера особенных скал.

Далее, несколько раз повторяется следующее:

- отец Федор вносит изменение в структуру скал:
 - если отец Федор собирается добавить путь длины l между вершинами a и b , выведите строку «`add abl`» ($1 \leq a, b \leq n, a \neq b, 0 \leq l \leq 10^{15}$)
 - если отец Федор собирается удалить путь длины l между вершинами a и b , выведите строку «`delete abl`» ($1 \leq a, b \leq n, a \neq b, 0 \leq l \leq 10^{15}$)
- программа жюри сообщает вашей программе одно целое число v — длину пути между скалой с отцом Федором и скалой с выходом
- если длина кратчайшего пути удовлетворяет отца Федора, выведите «`answer ok`» и завершите программу
- если отец Федор понял, что ему не выбраться со скал, выведите «`answer impossible`» и завершите программу

Отец Федор может сделать не более $5 \times m$ действий.

Пример

stdin	stdout
3 1 3	add 2 3 4
5 7	delete 2 3 4
2	add 2 3 3
2 3	answer ok
8	
10	
7	

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.