Задача A. Гонки на подах

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

Гонки на по́дах — популярный вид развлечений. Удовольствие от них способны получить и участники, и зрители, и букмекерские конторы, принимающие ставки на результат очередной гонки. При этом владельцы букмекерских контор периодически совершают некоторые незаконные действия, чтобы повлиять на итоговые позиции гонщиков. Например, можно испортить тормозную систему в некоторых подах, после чго гонщику придется ехать существенно медленнее обычной скорости ради сохранения контроля гонщика над машиной.

Выяснилось, что в последней гонке участвовали n гонщиков. У каждого из гонщиков на машине был написан номер — число от 1 до n, номера всех гонщиков различались. Также известно, что владельцы одной из букмекерских контор испортили тормозную систему во всех машинах, номера которых превосходили некоторое число k. Любая машина с испорченной тормозной системой будет ехать медленнее, чем любая машина с исправными тормозами. Соответственно, в протоколе с результатами гонки у любой машины с номером большим, чем k, место будет также больше, чем k.

Вы проводите расследование этого неприятного инцидента. В качестве первого шага расследования вы решили найти все возможные значения числа k, изучая только результаты гонки.

Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \le n \le 100\,000$) — количество гонщиков, участвовавших в соревновании. Вторая строка содержит n различных чисел a_i ($1 \le a_i \le n$) — протокол с результатами гонки, где a_i — номер машины, которая заняла i-ое место.

Формат выходного файла

В первой строке выходного файла выведите одно целое число c — количество возможных значений числа k. В следующей строке выведите c натуральных чисел, разделенных пробелами — возможные значения числа k. Все числа во второй строке должны быть различны и не должны превосходить n. Числа во второй строке должны быть упорядочены по возрастанию.

Примеры

race.in	race.out
6 2 1 4 3 6 5	3 2 4 6

Задача B. Склад Оби-Вана Кеноби

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

Оби-Ван Кеноби — один из последних представителей ордена рыцарей джедаев. Оби-Ван Кеноби родом с планеты Стьюджон, где абсолютно все жители чтут порядок, и Оби-Ван Кеноби — не исключение.

Во всех эпизодах «Звездных войн» Оби-Ван Кеноби орудует синим световым мечом, но мало кто знает, что на самом деле у него их много, причем все мечи Кеноби пронумерованы. Оби-Ван Кеноби хранит все свои мечи на очень длинном столе. Когда он хочет вооружиться, то он берет самый правый меч со стола, и идет по своим делам. Оби-Ван Кеноби орудует взятым мечом пока не потеряет или не сломает его.

Иногда кто-нибудь дарит Оби-Вану Кеноби новый меч, и тогда он просто кладет его на стол справа к тем, что уже лежат там. Но самое страшное случается тогда, когда к столу подходит мама Оби-Вана Кеноби. Мама тоже житель планеты Стьюджон, и поэтому она всегда возмущается, что мечи лежат неупорядоченно. Чтобы исправить это, она забирает всю левую половину мечей (если мечей на столе было нечетное количество, то мама забирает наибольшее целое число мечей, меньшее половины их общего количества) и начинает их подкладывать справа к оставшимся. Причем сначала она кладет меч, который изначально был самым левым, потом правее от него кладет второй меч и так далее.

Определите, в каком порядке лежат мечи у Оби-Вану Кеноби на данный момент. У каждого меча есть свой личный номер, который известен только Оби-Вану Кеноби. Ну и, конечно, вам.

Формат входного файла

В первой строке входного файла дано натуральное числа n — количество изменений, произошедших на столе ($1 \le n \le 10^6$). В следующих n строках заданы описания изменений в следующем формате:

- если строка начинается со слова `add`, то в той же строке через пробел находится число x ($1 \le x \le n$) — личный номер меча, подаренного Оби-Вану Кеноби. Гарантируется, что мечей с таким номером Оби-Вану Кеноби раньше не дарили и больше не подарят.
- если строка содержит единственное слово `take`, то это обозначает, что Оби-Ван Кеноби забрал самый правый меч со стола с собой. Если на столе не было мечей, то ничего не произошло.
- если же строка содержит единственное слово `mum!`, то это обозначает, что к столу подошла мама и навела порядок. Если на столе было меньше двух мечей, то ничего не произошло.

Изначально стол был пуст.

Формат выходного файла

В первой строке выходного файла выведите число k — количество мечей, лежащих на данный момент на столе у Оби-Вана Кеноби. В следующей строке выведите k чисел через пробел — личные номера мечей, лежащих на столе, в том порядке, в котором они на нем лежат (слева направо).

Примеры

kenobi.in	kenobi.out
8 add 1 add 2 add 4 add 3 add 5 add 8 take mum!	5 4 3 5 1 2

Пояснение

После выполнения операций `add` и `take` на столе будет лежать пять мечей в следующем порядке: 1 2 4 3 5. После чего придет мама Оби-Вана Кеноби, возьмет мечи с номерами 1 и 2, и положит их права от оставшихся трех мечей на столе.

Задача C. Покраска забора

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

Юному Энакину Скайуокеру магистр Йода поручил, на первый взгляд, простое задание. Ему необходимо было покрасить забор специальной краской. Задание состояло из n подзадач. Каждая из подзадач заключалась в покраске некоторого участка забора краской заданного цвета. Каждый цвет обозначался целым числом. Краска, которая была дана Энакину, как и любая другая, после того, как высыхала, меняла свой цвет. Но это происходило немного необычным образом.

Будем говорить, что два участка забора пересекаются, если они имеют не нулевую площадь пересечения. Разобьем все участки забора, на которые была нанесена краска, не некоторые множества. Вначале каждый отрезок отнесем к отдельному множеству. Если существуют пересекающиеся участки, которые относятся к различным множествам A и B, обьединим A и B. Будем делать так до тех пор, пока такие участки не перестанут существовать.

Теперь для каждого множества можно определить его цвет как среднее арифметическое всех цветов входящих в него участков. Зная эту величину для каждого множества, мы легко можем определить, какого цвета будет некоторая точка забора после высыхания краски. Будем говорить, что точка принадлежит множеству, если есть хотя бы один участок в множестве, который содержит в себе эту точку. Если точка забора не принадлежит ни одному множеству, она будет иметь цвет 0, иначе она будет иметь цвет соответствующего множества. Заметим, что точка не может принадлежать сразу двум множествам.

Пока Энакин занимался покраской забора, ему стало интересно, что же случится, если он закончит покраску после нанесения первых i участков. В частности, его заинтересовал такой вопрос «Точку какого наибольшего цвета можно будет найти на заборе после того, как краска высохнет?».

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \le n \le 10^5$) — количество участков забора, которые необходимо покрасить. В следующих n строках находится по три целых числа l, r, c ($0 \le l < r \le 10^9, 1 \le c \le 10^9$) — левая и правая границы i-го участка (левая граница включается, правая нет), а также цвет, в который необходмо покрасить соответствующий участок.

Формат выходного файла

В n строках необходимо вывести по одному вещественному числу с не менее чем тремя знаками после запятой. Число в i-ой строке должно обозначать максимальный цвет точки на стене, который можно было бы найти, если бы Энакин закончил покраску забора после выполнения первых i подзадач.

Примеры

painting.in	painting.out
6 0 1 1 2 5 2 3 7 3 6 8 4 9 11 5 7 10 6	1.00000 2.00000 2.50000 3.00000 5.00000 4.00000

Задача D. Отряд

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

Повстанцы снова атакают Империю на планете Хот, и Дарт Вейдер решил послать туда своих лучших клонов. Но вот незадача: звездолет всего лишь один, а отрядов для его управления два. Тогда он решил объединить эти отряды так, чтобы в получившемся отряде остались только лучшие клоны.

Оба отряда состоят из одинакового количества клонов. У каждого клона в отряде есть своя специальность, причем в одном отряде нет двух клонов с одинаковыми специальностями. Понятно, что набор специальностей клонов в первом отряде совпадает с набором специальностей во втором.

Чтобы выбрать лучших клонов, для каждого отряда была сделана специальная карточка со всеми оценками клонов, полученными ими когда-то в академии. Известно, что карточки делались следующим образом:

- из первого отряда вызывался очередной клон
- из второго отряда вызывался клон, владеющий той же специальностью, что и клон из первого отряда
- оценка клона из первого отряда дописывалась справа на карточку первого отряда, а оценка клона из второго отряда — на карточку второго отряда
- описанные действия совершались до тех пор, пока в отрядах оставались неучтенные клоны
- оценка клона — число от 0 до 9

После этого по каждой специальности был выбран тот клон, оценка которого не меньше, чем у клона с аналогичной специальностью. В конце на получившийся отряд была заведена аналогичная карточка. Известно, что при ее составлении порядок специальностей был тот же, что и при составлении карточек для двух исходных отрядов. Однако, эту карточку потеряли, и теперь вам необходимо помочь ее восстановить.

Формат входного файла

В двух строках записаны два числа одинаковой длины a и b ($1 \le a, b \le 10^{18}$). a — число, которое было записано на карточке первого отряда, а b — второго. Числа имеют одинаковую длину и не содержат ведущих нулей.

Формат выходного файла

В единственной строке выведите число, которое было написано на карточке отряда, полученного в результате описанного объединения.

Примеры

squad.in	squad.out
1234 4321	4334

Задача E. Парад победы

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

В честь уничтожения Звезды Смерти и победы над Империей, повстанцы устроили парад. В параде участвуют n человек. У каждого из них есть какое-то количество наград. Оказалось, что у каждых двух человек количество наград различно.

Организаторы парада ответственно отнеслись к его проведению. Участники парада по очереди выходят на площать и строятся в линию. При этом, очередной выходящий проходит мимо всех уже стоящих на площади. Проходя мимо обладающего бóльшим числом наград человека, проходящий обязан остановиться и отдать ему честь. На это уходит некоторое одинаковое, прописанное в уставе, время.

Во время подготовки все строго обговорили порядок, в котором будут стоять на площади. Но им стало известно, что на самом параде будет выставлена техника, которая загородит один из входов на площадь. К сожалению, ещё не известно будет ли загорожен левый или правый вход. Но, так как вся программа парада выверена до секунды, то, с какой стороны участники будут выходить на площадь, не должно сказаться на том, сколько времени им на это потребуется. Время, которое потребуется на то, чтобы выйти всем участникам, зависит только от количества раз, которое будет отдана честь.

Если будет закрыт правый выход, то люди будут выходить следующим образом: сначала тот, кто должен стоять левее всех, затем тот, кто должен стоять вторым, пройдя мимо уже стоящего первого, и так далее. Если же закрыт левый выход, то сначала выходит тот, кто будет стоять самым правым, затем тот, кто будет стоять вторым справа, проходя мимо уже стоя́щего правого, и так далее.

Интересными являются только те порядки построения, при которых время парада при выходе участников спарва и слева будет одинаковым. Вам требуется определить их количество.

Формат входного файла

В единственной строке входного файла записано единственное целое число n ($1 \le n \le 500$).

Формат выходного файла

Выведите единственное число — количество способов, по модулю $10^9 + 7$.

Примеры

victory.in	victory.out
3	0
4	6

Пояснение

Если количества наград у участников парада во втором примере равны числам 1, 2, 3 и 4, то нас интересуют следующие порядки построения:

- 1, 4, 3, 2
- 2, 3, 4, 1
- 2, 4, 1, 3
- 3, 1, 4, 2
- 3, 2, 1, 4

- $4, 1, 2, 3$

Задача F. Война клонов

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

Армия клонов, созданная на планете Камино, помогала джедаям сражаться против дроидов. Все изменилось, когда Палпатин отдал «Приказ 66»: без предупреждения клоны напали на джедаев. Но не все еще потеряно для Светлой стороны. Некоторые клоны не услышали приказ и продолжили сражаться на стороне джедаев. Поэтому война не закончилась, когда все джедаи были убиты. По приказу Палпатина все клоны выстроились в одну линию для финальной битвы.

Магистр Йода понял, что он должен научиться отличать имперских клонов и клонов Республики, которые остались верны джедаям. Оглядев поле, Йода присвоил каждому клону координату, соответствующую месту, где стоит клон. С помощью Силы магистр изменил расстановку клонов на поле так, что расстояние между любыми двумя клонами одной стороны стало четно, а между любыми двумя клонами вражеских сторон — нечетно. При этом никакие два клона не стоят на одной позиции. И началась война клонов...

Чтобы оценить шансы на победу, магистр Йода хочет знать количество пар клонов, которые могут сразиться между собой, то есть количество пар таких, что клоны в паре служат разным сторонам.

Формат входного файла

В первой строке входного файла задано число n ($1 \le n \le 100000$) — число клонов, участвующих в финальной битве. В следующей строке заданы n чисел a_i ($1 \le a_i \le 10^9$) — координаты клонов, все a_i различны.

Формат выходного файла

В выходной файл выведите единственное числов — количество пар клонов, которые могут сразиться между собой.

Примеры

versus.in	versus.out
5 1 2 3 4 5	6

Задача G. Шифровка

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

Лее Органе срочно нужна ваша помощь! Орден Джедаев недавно смог перехватить шифровку с планеты Татуин, предназначавшуюся, видимо, для Ордена Ситхов. В этой сводке должно содержаться очень много информации, в том числе и та, которая интересна Светлой Стороне. Однако данная шифровка может являться также и дезинформацией, чтобы запутать доблестных борцов со злом.

Однако, как сообщил перебежчик, достоверность послания зависит от числа `интересных` пар префиксов и суффиксов одинаковой длины. Префикс длины k может образовать `интересную` пару с суффиксом той же длины, если и префикс и суффикс содержат одинаковое, ненулевое количество гласных букв. Во всей вселенной звездных войн принято считать гласными следующие буквы: «`a`», «`o`», «`u`», «`i`», «`e`».

К сожалению, наши друзья пока не знают зависимость между достоверностью шифровки и количеством пар суффиксов и префиксов. Скоро они выяснят эту зависимость, а пока Лея просит помочь ей посчитать количество `интересных` пар префиксов и суффиксов.

Формат входного файла

Во входном файле в первой строке содержится целое число n ($1 \le n \le 20000$) — длина зашифрованной строки. Вторая строка содержит саму шифровку, состоящую только из строчных и прописных букв латинского алфавита.

Формат выходного файла

В выходном файле выведите одно число: количество таких положительных чисел k, что префикс и суффикс длины k содержат одинаковое, ненулевое количество гласных букв.

Примеры

encryption.in	encryption.out
8 GoodLuck	2

Пояснение

Напоминаем, что гласные прописные символы тоже считаются гласными.

Задача H. Спичрайтер Йоды

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

Всем известно, что у каждого важного общественного деятеля есть собственный спичрайтер — существо, помогающее подготовить публичную речь, сделать ее более выразительной и интересной. Глава Ордена джедаев магистр Йода не исключение.

На первый взгляд может показаться, что спичрайтеру Йоды приходится тяжелее других: все-таки речь магистра своеобразна и ее изучение требует серьезных усилий. На самом деле все несколько проще. Спичрайтеру Йоды достаточно сначала придумать речь для обычного человека, после чего поменять порядок слов в каждом предложении на обратный. В силу того, что алгоритм преобразования обычной речи в речь магистра Йоды достаточно однообразен, спичрайтер решил автоматизировать этот процесс и попросил вас написать программу, которая будет преобразовывать речь, составленную им для обычного человека в речь для Йоды.

Формат входного файла

В единственной строке входного файла задана речь, составленная спичрайтером. Речь состоит из предложений, отделенных друг от друга точками (точка ставится сразу после последнего слова в предложении).

Каждое предложение состоит из слов. Предложение содержит по крайней мере одно слово. Соседние слова разделены ровно одним пробелом. Слово — непустая последовательность строчных латинских букв. Строка не содержит лишних пробелов. Гарантируется, что строка не пуста и ее длина не превосходит 20000 символов.

Формат выходного файла

В одной строке выведите преобразованную для Йоды речь в соответствии с форматом входных данных. Следует строго соблюдать формат вывода речи, описанный во входных данных.

Примеры

yoda.in	yoda.out
`you should solve this problem. its easy.`	`problem this solve should you. easy its.`