Задача A. Accordeon

Имя входного файла:	`accordeon.in` или *стандартный ввод*
Имя выходного файла:	`accordeon.out` или *стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

> Баян (боян, байан и др.) — обозначение повторно опубликованной шутки или информации. При этом, в классическом понимании, информация должна быть повторно опубликована в том же самом источнике (например в том же самом форуме, или даже в том же самом разделе форума). Иными словами, репост (копирование информации из одного источника в другой), бояном в классическом понимании не является.
>
> Lurkmore

Фёдор администрирует юмористический сайт. Каждый день на сайт пользователи выкладывают сотни и тысячи анекдотов. В иные дни анекдотов выкладывается даже больше чем девять тысяч. К сожалению, не все анекдоты уникальны, среди них часто встречаются «баяны». «Баяном» Фёдор называет анекдот, который совпадает с каким-нибудь анекдотом, уже выложенным на сайт. Однако пользователи зачастую заносят анекдоты в базу вручную, по памяти, потому в одном и том же анекдоте они могут наставить лишних пробелов или пропустить какие-то знаки пунктуации.

Более строго, перед сравнением Фёдор с каждым анекдотом проделывает следующие операции:

1. Все знаки препинания заменяются на пробелы.
2. Все заглавные латинские буквы заменяются на соответствующие им строчные.
3. Все последовательно идущие пробелы заменяются на один пробел.

Фёдор считает два анекдота одинаковыми, если после выполнения этих операций они совпадают. При этом пробелы в начале и конце каждого анекдота Фёдор игнорирует. Однако он устал уже проверять анекдоты вручную и просит Вас написать программу, которая ему поможет.

Формат входного файла

В первой строке входного файла записан первый анекдот, во второй строке — второй. Каждый анекдот - это строка не более чем из 10^4 символов: заглавных и строчных латинских букв, пробелов и знаков препинания (`.,:;!?`).

Формат выходного файла

Выведите `"YES"`, если анекдоты из входного файла совпадают (в понимании Фёдора) и `"NO"` иначе.

Примеры

accordeon.in или стандартный ввод
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once. Am i god?
accordeon.out или стандартный вывод
YES

accordeon.in или стандартный ввод
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once. AM I a GOD?
accordeon.out или стандартный вывод
NO

accordeon.in или стандартный ввод
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once AM I GOD
accordeon.out или стандартный вывод
YES

accordeon.in или стандартный ввод
I successfully stopped a print job once. AM I GOD? I successfully stopped a print job once . AM I GOD?
accordeon.out или стандартный вывод
YES

Задача B. Cards

Имя входного файла:	`cards.in` или *стандартный ввод*
Имя выходного файла:	`cards.out` или *стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

В одном IT-Парке государственного университета города P поставили турникет для учета входящих и выходящих сотрудников. Каждый работник имеет ключ-карту.

На столе у начальник охраны IT-парка лежат N прямоугольных ключ-карт, каждая из которых имеет длину L и ширину H. Начальник охраны пытается составить из них прямоугольник наименьшего периметра, использовав все карты.

Ваша задача — найти этот самый периметр

Формат входного файла

В первой строке входного файла задано одно целое число T — количество тестовых примеров ($1 \le T \le 5$)

Каждый тестовый пример состоит из трёх целых положительных чисел, не превосходящих 1000 — количества ключ-карт N, длины ключ-карты W и её ширины H.

Формат выходного файла

Для каждого тестового примера выведите одно целое число — минимальный периметр получившегося прямоугольника.

Примеры

cards.in или стандартный ввод	cards.out или стандартный вывод
3 3 30 40 7 30 40 4 40 30	260 380 280

Задача C. Clinic Troubles

Имя входного файла:	`clinic.in` или *стандартный ввод*
Имя выходного файла:	`clinic.out` или *стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Ваш друг — главный врач маленькой, но очень престижной клиники. Его клиника состоит из n палат, соединённых $n-1$ коридором. При этом клиника, естественно, представляет собой связное здание, то есть из любой палаты можно дойти до любой другой палаты, используя коридоры.

В начале i–го коридора стоит коробка, в которой лежит c_i пар одноразовых бахил. Каждый пациент, входя в коридор, надевает одну пару бахил и только потом идёт по коридору. Пройдя коридор, больной выкидывает бахилы. Если в коридоре не осталось бахил, то по этому коридору больше не пройдёт ни один больной. В каждой палате находится p_i больных, ожидающих осмотра. Главврач вызывает на осмотр сразу всех пациентов палаты разом, при этом пациенты начинают идти из своей палаты в направлении палаты номер 1 (в ней находится главврач). Если в каком-то коридоре на их пути закончились бахилы, то оставшиеся пациенты останутся в начале этого коридора и больше никуда не пойдут. К сожалению, за рабочий день главврач может осмотреть пациентов из не более чем k различных палат. Помогите ему выбрать осматриваемые палаты таким образом, чтобы он осмотрел как можно больше пациентов.

Формат входного файла

В первой строке входного файла дано два целых числа n и k — количество палат в клинике и максимальное число палат, которые может осмотреть доктор соответственно ($1 \le k \le n \le 250$). Во второй строке дано n целых чисел, p_i ($0 \le p_i \le 10^6$) — количество больных в i–ой палате. В следующих $n-1$ строках дано описание коридоров в больнице. Описание коридора — три целых числа u, v, c ($1 \le u < v \le n, 0 \le c \le 10^6$). Они означают, что в больнице есть коридор из палаты с номером v в палату с номером u, и в начале этого коридора стоит коробка, в которой лежит c пар бахил.

Формат выходного файла

В первую строку выходного файла выведите одно число — максимальное количество больных, которых может осмотреть главврач. В следующую строку выходного файла выведите число q — количество палат, которые главврач вызовет на осмотр. В следующую строку выведите q чисел — номера палат, которые главврач вызовет на осмотр. Если оптимальных решений несколько, то выведите любое.

Примеры

clinic.in или стандартный ввод	clinic.out или стандартный вывод
4 1 0 10 5 5 1 2 1 1 3 5 1 4 5	5 1 3
4 2 0 0 5 5 1 2 7 2 3 5 2 4 5	7 2 3 4
5 3 10 10 10 10 10 1 2 1 2 3 1 2 4 1 2 5 1	11 2 1 2

Note

В первом примере доктору невыгодно вызывать больных из второй палаты, поскольку оттуда до доктора сможет дойти лишь один больной (остальные так и останутся у входа в коридор из второй палаты в первую).

Во втором примере больные из палат 3 и 4 дойдут по коридорам до палаты 2, но в палату 1 сможет пройти только 7 больных из-за нехватки бахил.

В третьем примере доктору нет смысла вызывать больных из ещё какой-либо палаты, поскольку им заведомо не хватит бахил для перехода из палаты 2 в палату 1 (однако вызов больных, которые заведомо не дойдут до доктора, ошибкой в этой задаче не считается).

Задача D. Parentheses

Имя входного файла:	`parentheses.in` или *стандартный ввод*
Имя выходного файла:	`parentheses.out` или *стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Вместо того, чтобы готовиться к урокам или играть в снежки с друзьями, Вася решил посидеть в интернете и поиграть в браузерные игры. Например, в следующую.

Даётся строка, состоящая из маленьких английских букв и символов '(' и ')' (ASCII коды 40 и 41, соответственно). Необходимо с помощью операции **удаления** избавиться от всех скобок. При этом оставшаяся строка должна иметь наибольшую возможную длину.

Операция удаления состоит из выбора некоторой подстроки, которая начинается с открывающей круглой скобки и заканчивается закрывающей, и удаления этой подстроки. При этом оставшиеся части строки склеиваются.

Рассмотрим пример: $s = abac(a)ba(mn(o)go)eda$.

1. Выберем подстроку (a) и удалим. Останется $abacba(mn(o)go)eda$.
2. Выберем подстроку (o) и удалим. Останется $abacba(mngo)eda$.
3. Выберем подстроку $(mngo)$ и удалим. Останется $abacbaeda$.

Также в этом примере можно удалить, например, подстроку $(a)ba(mn(o)$, но в таком случае оставшуюся закрывающую скобку было бы невозможно удалить. Можно сразу удалить подстроку $(a)ba(mn(o)go)$, но в этом случае осталось бы $abaceda$ — строка не является оптимальной.

Помогите определить наилучшую стратегию и узнать, как следует делать ходы, чтобы осталось максимальное количество букв после удаления всех скобок.

Формат входного файла

В единственной строке файла содержится строка s длиной не более 10^5 символов, которая состоит из маленьких букв английского алфавита и символов '(' и ')'. Гарантируется, что всегда существует способ удаления, при котором остается только строка, состоящая только из символов английского алфавита.

Формат выходного файла

Выведите в единственной строке файла исходную строку, в которой символы, которые будут удалены, заменены на символы '*'.

Примеры

`parentheses.in` или стандартный ввод	`parentheses.out` или стандартный вывод
`(i)nt)`	`******`
`zanknvn()(t()l`	`zanknvn******l`

Задача E. Parking

Имя входного файла:	`parking.in` или *стандартный ввод*
Имя выходного файла:	`parking.out` или *стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

В центре города П. построили большой и красивый торговый центр. Его парковка состоит из N уровней. На i-м уровне есть a_i парковочных мест. При въезде на парковку водителю выдаётся талон, в котором указан минимальный номер уровня, на котором есть свободные места. Получив подобный талон, водитель отправляется на указанный уровень и занимает там место. По окончании визита в торговый центр, покупатель уезжает, освобождая парковочное место.

Естественно, в течение дня на парковку приезжает и уезжает большое количество автомобилей, поэтому для выдачи талонов на парковку хозяева центра хотят использовать программу. От вас требуется такую программу реализовать.

Формат входного файла

В первой строке входного файла находится единственное число N — количество уровней на парковке ($1 \le N \le 10^5$). Во второй строке находится N чисел a_i — количество парковочных мест на каждом уровне ($1 \le a_i \le 1000$).

В третьей строке входного файла находится число M — количество событий, произошедших на парковке за день ($1 \le M \le 10^5$). В четвёртой строке находится M чисел b_i, описывающих события. Если $b_i = 0$, значит на парковку приехал очередной автомобиль. Если же $b_i = x > 0$, значит, с уровня номер x уехал один из автомобилей, расположенных там. Конечно, во втором случае гарантируется, что $x \le N$ и что в этот момент на уровне x находится хотя бы один автомобиль.

Известно, что мест для парковки автомобилей заведомо хватает, то есть при каждом появлении нового автомобиля для него есть хотя бы одно место.

Формат выходного файла

Выведите столько чисел, сколько раз событие «приехал автомобиль» встретилось во входном файле. Для каждого такого события выведите единственное число — номер на талоне, выдаваемом водителю этого автомобиля. Напоминаем, что этот номер должен быть равен минимальному номеру уровня, на котором есть свободные парковочные места.

Примеры

parking.in или стандартный ввод	parking.out или стандартный вывод
4 2 2 2 2 6 0 0 0 0 1 0	1 1 2 2 1
5 1 1 1 1 1 11 0 0 0 0 0 5 1 3 0 0 0	1 2 3 4 5 1 3 5

Note

Решения, работающие при $1 \le N, M \le 1000$ будут оцениваться из 60 баллов.

В первом примере первые два автомобиля помещаются на первый уровень парковки. Два следующих автомобиля нельзя поместить на первый уровень, поскольку все места там заняты. Таким образом, эти автомобили помещаются на второй уровень. Далее, один из автомобилей с первого уровня уезжает и следующий приехавший автомобиль может занять его место.

Задача F. Rake

Имя входного файла:	`rake.in` или *стандартный ввод*
Имя выходного файла:	`rake.out` или *стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мебибайт

Петя приехал на дачу к своему деду. Дедушка в молодости воевал, потому очень ответственно подошёл к защите своей дачи от нежелательных посетителей. Во-первых, тропинка к дому засажена по обе стороны особо колючими кустами, поэтому свернуть с неё никак не получится. А во-вторых, на тропинке разложены N различных видов замаскированных грабель в расчёте на то, что незваные гости будут наступать на них и получать этими самыми граблями по лбу.

Петя помнит, что грабель i-го вида у деда имеется b_i штук. Если наступить на грабли i-го вида a_i раз, то можно наконец получить необходмый опыт для преодоления таких грабель в следующий раз и более ни разу ими по лбу не получать. Помимо этого Петя умеет перепрыгивать грабли, но может это сделать не более чем K раз. При этом перепрыгивание грабель хоть и позволяет избежать удара по лбу, но не даёт опыта в обхождении таких грабель. Следовательно, чтобы научиться обходить грабли i-го вида, Пете придётся обязательно получить такими граблями a_i раз по лбу.

Прежде чем предпринять отчаянную попытку по штурму дачной тропинки, Петя попросил Вас помочь ему в определении минимально возможного числа ударов граблями по лбу, необходимого для прохождения полосы препятствий.

Формат входного файла

В первой строке входного файла находятся два числа N и K — количество видов грабель, разложенных заботливым дедом и количество прыжков, которое может совершить Петя ($1 \le N \le 10^4, 1 \le K \le 10^3$).

Далее следует N строк, по два числа в каждой — a_i и b_i, количество раз, которое требуется получить граблями i-го вида по лбу, чтобы научиться их обходить и количество грабель i-го вида, лежащих на пути до дачи ($1 \le a_i \le b_i \le 10^3$).

Формат выходного файла

Выведите единственное число — минимально возможное количество ударов по лбу, которое получит Петя, добравшись до дачного домика.

Примеры

rake.in или стандартный ввод	rake.out или стандартный вывод
2 5 3 5 1 5	1
3 3 5 10 5 10 5 10	15

Note

В первом примере Пете нужно перепрыгнуть все грабли первого вида, потратив при этом все прыжки. После этого он один раз получит по лбу граблями второго вида, в результате чего научится их обходить и минует все оставшиеся грабли.

Во втором примере Петя в любом случае получит пять раз по лбу граблями каждого вида, поскольку даже если он перепрыгнет грабли какого-нибудь вида три раза, после этого он всё равно из оставшихся семи грабель такого вида пятью получит по лбу (и только потом научится эти грабли обходить).