Задача A. 01

Имя входного файла:	`01.in`
Имя выходного файла:	`01.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Задана строка длины N, состоящая из символов `0` и `1`. Необходимо с помощью последовательности обменов привести ее к такому виду, что сначала идут все символы `0`, а затем идут все символы `1`. Каждый обмен можно задать парой различных чисел i, j $(1 \le i < j \le N)$. За каждый такой обмен назначается штраф, равный $\sqrt{j-i}$.

Какой минимальный суммарный штраф придется уплатить за выполнение поставленной задачи?

Формат входного файла

Во входном файле записана заданная строка. Длина строки не менее 1 символа и не превосходит 4000 символов.

Формат выходного файла

Выведите искомый минимальный штраф. Не допускается превышение одновременно абсолютной и относительной погрешности более чем на 10^{-9}.

Пример

01.in	**01.out**
`100`	`1.414213562373`

Задача B. Кипяток

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

Как хорошо выпить чай в купе во время поездки! Особенно хорошо это делать не в одиночку. В купе собралось N человек, и каждый пришел со своей кружкой. Кружка i-го человека вмещает A_i миллилитров воды. Вам, как проигравшему в интеллектуальную игру «О, Макс!», надо наполнить кипятком все кружки. Кипяток можно получить в конце вагона, но туда можно идти только с одной кружкой. Эта кружка может быть как пустой, так и частично наполненной. Она доливается кипятком (не обязательно полностью), и вы возвращаетесь назад в купе. Там вы можете произвольным образом переливать воду из кружки в кружку. Так как поезд оснащен новым цифровым паровым котлом, кипяток в кружку доливается всегда порциями в целое число миллилитров. За одну ходку за кипятком вы тратите энергию равную $W^2 + M$, где W — это масса кипятка, долитого в кружку (вода, которая уже была в ней, не учитывается, так как успела уже остыть), а M — это масса вашего тела.

Какое наименьшее количество энергии необходимо потратить, чтобы наполнить водой все кружки в купе?

Формат входного файла

В первой строке входного файла записаны два целых числа N, M ($1 \le N \le 50000$, $0 \le M \le 10^9$). Далее во входном файле записана последовательность целых чисел $A_1, A_2, \ldots, A_N$ ($1 \le A_i \le 10^5$).

Формат выходного файла

Выведите искомую величину в виде целого числа без лидирующих нулей.

Пример

boiling.in	boiling.out
2 5 2 2	18
2 12 5 1	42

Задача C. Посадка в поезд

Имя входного файла:	`boarding.in`
Имя выходного файла:	`boarding.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Цифровой поезд нового поколения состоит из вагонов, содержащих по N мест для пассажиров. Все места расположены вдоль вагона и пронумерованы от 1 до N. Вход в вагон расположен левее места 1, а места $1, 2, \ldots, N$ расположены правее от входа в соответствующем порядке. N пассажиров готовятся сесть в поезд. Каждый заходящий в вагон характеризуется номером места A_i и своей массой B_i. Когда пассажир идет по вагону от входа до своего места, некоторые пассажиры, которые сели ранее, мешают ему пройти. Пассажир испытывает неудобство, каждый раз проходя мимо человека массы большей, чем он сам. Суммарным неудобством пассажира при посадке называется количество раз, когда он испытывал неудобство при движении к своему месту от входа в вагон.

Ваша задача по заданному порядку посадки пассажиров найти суммарное неудобство каждого.

Формат входного файла

Входной файл состоит из одного или нескольких наборов входных данных. Каждый набор начинается с целого числа N ($1 \le N \le 16000$) — количества мест (пассажиров). Далее набор входных данных содержит пары целых чисел A_i, B_i ($1 \le A_i \le N$, $1 \le B_i \le 10^9$). Все числа A_i и B_i различны между собой. То есть номера мест пассажиров образуют перестановку. Пассажиры садятся в поезд имеенно в том порядке, как они заданы во входном файле. Количество наборов входных данных не превосходит 5.

Формат выходного файла

Выведите ответ на каждый набор входных данных. Каждый ответ должен состоять из последовательности N целых чисел $P_1, P_2, \ldots, P_N$, где P_i суммарное неудобство i-ого пассажира.

Пример

boarding.in	boarding.out
3 1 2 2 3 3 1 2 1 1 2 2	0 0 2 0 0

Задача D. Обратная префикс-функция

Имя входного файла:	`inverse.in`
Имя выходного файла:	`inverse.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k|0 \le k < i, S[1..k] = S[i-k+1..i]\}$ для все $1 \le i \le N$, где N — длина строки. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Ваша задача по заданной префикс-функции восстановить строку. В качестве символов строки разрешается использование M первых строчных букв латинского алфавита.

Формат входного файла

Входной файл состоит из одного или более набора входных данных.

В первой строке каждого набора записаны два целых числа N, M ($N \ge 1$, $1 \le M \le 26$). Во второй строке записана последовательность целых чисел $\pi[1], \pi[2], \ldots, \pi[N]$. Все числа в последовательности целые неотрицательные, не превосходящие 10^6.

Сумма значений N по всем наборам не превосходит 10^6, количество наборов входных данных не превосходит 10^5.

Формат выходного файла

Выведите в первую строку выходного файла `YES` если существует искомое слово и `NO` в противном случае. В случае положительного ответа выведите во вторую строку выходного файла выведите искомое слово. Если решений несколько, выведите любое. Разделяйте ответы на каждый набор входных данных пустой строкой.

Пример

inverse.in	**inverse.out**
8 3 0 0 1 0 1 2 3 1 1 1 10	YES abacabaa NO

Задача E. Корпорация

Имя входного файла:	`corporation.in`
Имя выходного файла:	`corporation.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	64 мегабайта

Корпорация, название которой по известным причинам оглашено не будет, имеет древесную структуру. Во главе компании стоит Президент, которому подчиняются (вероятно, косвенно, то есть по длинной цепочке) все остальные служащие. Каждый сотрудник (разумеется, кроме Президента) имеет одного непосредственного начальника. Каждый служащий (кроме служащих нижнего звена) имеет подчиненных, которых он нумерует от 1. Возможно, что подчиненный всего один. Всем сотрудникам, даже Президенту, присвоен индивидуальный номер от 1 до N.

Структура корпорации является корпоративной тайной и строго охраняется. Однако, на сайте компании доступна следующая информация. Известно, что в процессе «переписи корпорации» был создан один специальный бланк, который передавался от Президента. При получении этого бланка сотрудник передавал его по очереди всем своим подчиненным, затем расписывался сам и возвращал бланк своему непосредственному начальнику. Так бланк обошел всю корпорацию и вернулся к Президенту, который тоже поставил на нем свою подпись. На сайте корпорации доступен этот бланк, расшифровав который выяснили, что все сотрудники в нем расписались по порядку — от сотрудника с индивидуальным номером 1 до сотрудника с номером N. Таким образом, например, можно утверждать, что Президент имеет номер N.

Ваша задача восстановить структуру корпорации. Конечно, при $N > 2$ однозначно это сделать невозможно. Найдите все возможные структуры. Вам задана последовательность $P_1, P_2, \ldots, P_K$, надо вывести номера непосредственных начальников для служащих с номерами $P_1, P_2, \ldots, P_K$ ($1 \le P_1, P_2, \ldots, P_K < N$) для каждой возможной структуры корпорации.

Формат входного файла

Входной файл содержит целое число N ($2 \le N \le 15$). Во второй строке записано число K ($1 \le K < N$). В третьей строке записана последовательность различных целых чисел $P_1, P_2, \ldots, P_K$. Гарантируется, что входные данные таковы, что размер вывода не превосходит 12 МБ.

Формат выходного файла

Ответ для каждого возможного варианта структуры корпорации выводите на отдельной строке. Каждый ответ состоит из последовательности K чисел — номеров непосредственных начальников для служащих с номерами $P_1, P_2, \ldots, P_K$.

Пример

corporation.in	corporation.out
4 3 2 1 3	4 4 4 4 2 4 3 2 4 3 4 4 3 3 4

Задача F. Игра со словами

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

Девочка Катя любит разные игры со словами. Недавно она придумала новую игру. Увидев где-либо написанное слово, она пытается придумать предложение, содержащее это слово в себе. Буквы увиденного слова должны идти в предложении по порядку, но не обязательно подряд. Например, увидев слово «LTIBE», она придумывает предложение «LET IT BE». Разумеется, Катя ищет кратчайшее предложение.

Ваша задача помочь Кате, автоматизировав игру.

Формат входного файла

В первой строке содержится увиденное Катей слово. Слово состоит из заглавных латинских букв. Его длина не менее 1 символа и не более 500. Во второй строке входного файла записано целое число N ($1 \le N \le 500$) — количество слов в словарном запасе Кати. Далее в N строках записаны слова. Слова состоят из заглавных латинских букв, содержат не менее 1 и не более 50 букв. Слова в наборе могут совпадать. В искомом предложении слово может использоваться любое количество раз, главное, чтобы оно содержалось в списке.

Формат выходного файла

Выведите искомое предложение. Слова в предложении разделяйте пробелами. Длиной предложения называется количество символов (включая пробелы) использующихся при его записи. Переведите в нижний регистр те символы, которые образуют заданное слово. Если решений несколько, выведите любое. Если решения не существует, выведите в выходной файл символ «*».

Пример

game.in	game.out
LTIBE 5 IT THE LET BEE BE	lEt iT be
TBONTOBE 4 OR BE TO NOT	tO bE oR nOt To be

Задача G. Планета Круг

Имя входного файла:	`planet.in`
Имя выходного файла:	`planet.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Страна Флатляндия расположена на планете Круг в двумерном мире. Система координат введена таким образом, что центр планеты совпадает с началом системы координат. Радиус Круга равен R ($R \geq 1$). Планета вращается вокруг двух солнц — Зеленого и Красного. Оба солнца чрезвычайно малы по размерам и геометрически эквивалентны точкам. Зеленое Солнце в настоящий момент расположено в точке с координатами (x_g, y_g), а Красное — (x_r, y_r). Весь двумерный мир рассекают две Линии — первая из них не пропускает лучи Зеленого Солнца, а вторая — Красного. Первая Линия проходит через различные точки $(x_{1,1}, y_{1,1})$ и $(x_{1,2}, y_{1,2})$. Вторая — через различные точки $(x_{2,1}, y_{2,1})$ и $(x_{2,2}, y_{2,2})$.

Ваша задача — посчитать долю площади планеты Круг, которая освещается как Зеленым Солнцем, так и Красным.

Формат входного файла

Входной файл содержит один или более набор входных данных. Каждый набор содержит вещественные числа R, x_g, y_g, x_r, y_r, $x_{1,1}$, $y_{1,1}$, $x_{1,2}$, $y_{1,2}$, $x_{2,1}$, $y_{2,1}$, $x_{2,2}$, $y_{2,2}$. Все числа по абсолютной величине не превосходят 10^4. Количество наборов не превосходит 10^4.

Формат выходного файла

Выведите искомую долю. Не допускается превышение одновременно абсолютной и относительной погрешности более чем на 10^{-9}.

Пример

planet.in	planet.out
10 1 0 0 1 0 0 0 1 0 0 1 0 10 100 0 -100 0 -1 -1 1 -1 -1 1 1 1	0.2500 0.12711142843

В первом примере освещенная площадь планеты составляет примерно 78.5398163 кв. ед., а площадь всей планеты — 314.1592653589 кв. ед.

Задача H. Сортировка слов

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

Одна из новых возможностей текстового редактора «World XP» — это сортировка слов в предложении. Выход новой бета-версии редактора должен состоятся не позднее, чем через пять часов, а заявленная функция еще не реализована.

Требуется написать программу, осуществляющую сортировку слов в предложении. При этом все символы, отличные от букв, должны сохранится и не поменять своего положения относительно вхождений слов. Для упрощения при подаче входных данных на вход вашей программы все такие символы будут заменены на символ «`.`» (точка). Таким образом символ «`.`» имеет смысл разделителя между словами. Например, строка «`..aba.a..ba`» после сортировки пример вид «`..a.aba..ba`», а строка «`c..bb.a`» примет вид «`a..bb.c`». Слова следует сортировать лексикографически, как в словаре.

Формат входного файла

Входной файл содержит единственную строку, содержащую только прописные латинские буквы и символ «`.`». Слова могут разделяться любым количеством символов «`.`», строка может как начинаться так и заканчиваться последовательностью точек. Длина заданной строки не менее 1 символа и не превосходит 10^5 символов.

Формат выходного файла

В выходной файл выведите строку после сортировки слов в ней.

Пример

sort.in	sort.out
`..aba.a..ba`	`..a.aba..ba`
`c..bb.a`	`a..bb.c`

Задача I. Средний налог

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

Страна Флатляндия разделилась на «желтых» и «зеленых». Каждый город теперь поддерживает одну из двух партий. Все города стали «желтыми» или «зелеными». Каждая дорога соединяет пару различных городов, все дороги двунаправлены. Любые два города соединены не более чем одной дорогой. Теперь купля-продажа товара возможна только между одноцветными городами. Транспортировка так же осуществляется между одноцветными городами, но так как, возможно, не все пары одноцветных городов соединены дорогами провоз осуществляется через промежуточные города. При провозе товара через город цвета, отличного от цвета городов начала и конца пути, товар облагается налогом в 3 флатляндских евро. Так же необходимо уплачивать 1 флатляндский евро за проезд по любой дороге.

Ваша задача определить средний налог при провозе товара между каждой парой одноцветных городов. Разумеется, если путей между данной парой несколько, то выбирается путь с минимальным налогом. Гарантируется, что по дорогам можно проехать из любого города в любой другой.

Формат входного файла

В первой строке записаны два целых числа N, M ($3 \le N \le 1000$, $N - 1 \le M \le 5000$), где N — количество городов в стране, M - количество дорог. Пусть все города пронумерованы от 1 до N. Во второй строке содержится строка из N символов. i-ый символ строки равен Y если город поддерживает «желтых» и G если «зеленых». В следующих M строках записаны дороги номерами соединяемых городов.

Формат выходного файла

Выведите искомое значение. Не допускается превышение одновременно абсолютной и относительной погрешности более чем на 10^{-9}.

Пример

tax.in	**tax.out**
5 5 YYGYG 1 2 5 1 3 4 3 2 5 4	4.00000000