

Задача А. Напёрсточник

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Выиграть у напёрсточника — дело непростое! У него имеется n напёрстков, расставленных в ряд. Игра происходит следующим образом: сперва напёрсточник кладёт монетку под k -ый напёрсток, после чего начинает менять некоторые пары напёрстков. Известно, что при обмене пары напёрстков монетка, находящаяся под одним из них, перемещается вместе с ним.

К счастью, известна стратегия напёрсточника. Обмены происходят по следующему алгоритму. Рассмотрим некоторый набор напёрстков S :

1. если набор содержит два напёрстка, то обменяем их местами;
2. иначе рассмотрим все множества напёрстков в лексикографическом порядке, состоящие из $|S| - 1$ напёрстков, где $|S|$ — количество напёрстков в наборе S , и перейдём к шагу 1. Например, если S было $\{1, 2, 3, 4\}$, то в результате последовательно будут рассмотрены следующие наборы: $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$.

Требуется предсказать, где окажется монетка после махинаций напёрсточника.

Формат входного файла

Первая строка содержит числа n и k ($2 \leq n \leq 10^{18}, 1 \leq k \leq n$).

Формат выходного файла

Единственное число — номер напёрстка под которым окажется монетка по итогам действий напёрсточника.

Примеры

<code>stdin</code>	<code>stdout</code>
4 2	4
3 1	3

Задача В. Изогнутый палиндром

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Даны две строки: A и B . Длины обеих строк одинаковы и равны n .

Изогнутой называется такая строка, символы которой образованы конкатенацией подстроки A с символа i по символ j и подстроки B с символа j по символ k , $1 \leq i \leq j \leq k$.

Изогнутый палиндром — это либо изогнутая строка, являющаяся палиндромом, либо подстрока-палиндром любой из строк A или B .

Необходимо найти любой изогнутый палиндром максимальной длины и вывести его.

Формат входного файла

Первая строка содержит число n ($1 \leq n \leq 10^5$). Следующие две строки содержат слова A и B , состоящие только из заглавных латинских букв.

Формат выходного файла

В первой строке вывести единственное число — длину максимального изогнутого палиндрома. Последующие две строки должны содержать слова A и B , в которых все символы, не входящие в максимальный изогнутый палиндром, заменены символом "." (точка)

Примеры

stdin	stdout
5 ABCDE BAECB	5 .BC.. ..ECB

Задача С. Бег зайца через поля

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Однажды мой дед в Мичигане приручил зайца. Настоящего зайца с длинными ушами. А потом он удрал, и по нему начали палить охотники... Они сбегались отовсюду... Целых трое суток они гнались за ним через поля... И всякий раз он находил новую уловку и бежал быстрее, еще быстрее... (Жапризо С. «Бег зайца через поля».)

...Так вот, они загнали зайца на ранчо моего деда, огражденное забором в форме выпуклого многоугольника $A_1A_2 \dots A_n$. Заяц выбежал из внутренней точки участка и побежал в сторону границы A_1A_2 , надеясь спрятаться в кустах. Достигнув ограды A_1A_2 , он пробежал вдоль неё расстояние d_1 и повернул в сторону ограды A_2A_3 , где были другие заросли, затем пробежал расстояние d_2 вдоль нее, и так далее. В конце пути заяц перебежал к ограде A_nA_1 и пробежал расстояние d_n вдоль A_nA_1 . После этого он вернулся в исходную точку A , где его радостно принял в свои руки мой дед, окончательно запутав охотников, которые гнались за ним.

Вам необходимо вычислить длину кратчайшего пути зайца, который начинается и заканчивается в точке A .

Формат входного файла

В первой строке содержится одно натуральное число n — количество вершин забора ($3 \leq n \leq 200$). В каждой i -ой из n следующих строк записаны два числа — координаты вершин забора в порядке его обхода против часовой стрелки. Следующая строка содержит n целых чисел — расстояния d_1, d_2, \dots, d_n , причем $0 \leq d_i < l_i$ (l_i — длина стороны A_iA_{i+1} , l_n — длина стороны A_nA_1). В следующей строке записаны еще два числа — координаты точки A . Координаты всех точек — целые числа, не превосходящие по модулю 10^6 . Все числа разделены пробелами. Гарантируется, что кратчайший путь не проходит через вершины многоугольника.

Формат выходного файла

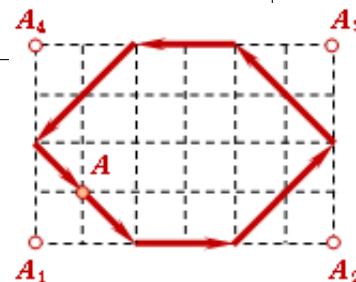
Выведите длину кратчайшего пути зайца с точностью не менее 10^{-5} .

Примеры

stdin	stdout
4 0 0 6 0 6 4 0 4 2 0 2 0 1 1	15.313708

Note

На рисунке изображен кратчайший путь зайца, содержащий два отрезка длины 2, расположенных на сторонах A_1A_2 и A_3A_4 участка $A_1A_2A_3A_4$. Его длина равна $8\sqrt{2} + 4 = 15.313708\dots$



Задача D. Лента Мёбиуса

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

И понеслись вдоль дороги, подгоняемые ветром, ленты Мёбиуса и замысловатые фигуры Лиссажу. Из сборника „Физики шутят“.

Кто не знает, что такое лента Мёбиуса? Вы не знаете? Это очень просто. Берем за концы ленту прямоугольной формы. Правый конец ленты поворачиваем на 180 градусов и склеиваем концы ленты. Получилась лента Мёбиуса. Вот такая.



Отличительная особенность ленты Мёбиуса состоит в том, что это односторонняя поверхность в отличие от простого прямоугольника. Или, пользуясь научным определением, лента Мёбиуса — это топологический объект, попасть из одной точки поверхности которого в любую другую можно, не пересекая края. Эту ленту придумал немецкий математик и астроном Август Фердинанд Мёбиус, профессор университета города Лейпциг (Германия), наблюдая за горничной, которая случайно неправильно одела на шею свой платок. Кстати, некоторые учёные считают, что наша Вселенная тоже устроена по принципу ленты Мёбиуса.

Лента Мёбиуса разбита на квадраты. Некоторые квадраты непроходимы, в них записан знак `*`. Перейти в соседний квадрат можно, если в нем нет звездочки и он имеет с текущим общую сторону.

Необходимо найти длину пути между двумя квадратами. Расстояние между двумя соседними квадратами считается равным единице.

Формат входного файла

В первой строке записаны два числа N и M ($1 < N \leq 1000, 1 < M \leq 1000$) — размеры исходного прямоугольника $PQRS$.

Каждая из последующих $2N$ строк содержит по M символов. Допустимыми символами являются `'.'` (точка, обозначает обычный квадрат), `'*'` (звездочка, обозначает непроходимый квадрат), `'D'` (обозначает стартовый квадрат) и `'K'` (обозначает финишный квадрат), причём два последних символа встречаются ровно по одному разу.

Первые N строк задают одну из сторон ленты, при этом точке P соответствует левый нижний угол, точке Q — правый нижний, точке R — правый верхний и точке S — левый верхний. Следующие N строк задают обратную сторону ленты, при этом лента ориентирована так, что точке P соответствует правый нижний угол, а точке S — правый верхний.

При построении ленты точка P склеивается с точкой R , а точка Q — с точкой S .

Формат выходного файла

Выведите минимальную длину пути между стартовым и финишным квадратами.

Примеры

stdin	stdout
2 5 D... ***** ***** ..K..	3

Задача Е. Шайбу!

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Прямоугольное поле для игры в шайбу имеет размеры $N \times M$. В исходном положении круглая шайба, имеющая радиус R , прижата к левому борту и её центр находится в точке с координатами $(R, \frac{N}{2})$. Суть игры состоит в том, чтобы провести шайбу в конечное положение, где её центр будет находиться в точке B с координатами $(M - R, \frac{N}{2})$. При этом центр шайбы должен пройти путь минимально возможной длины.

Игра была относительно простой, пока Федерация шайбистов не ввела новое правило. Теперь на поле установлены K перегородок, параллельных стороне длиной N . В каждой перегородке может быть несколько проемов, ширина которых всегда равна диаметру шайбы. Теперь игра сильно усложнилась, и найти кратчайший путь шайбы без использования высоких технологий почти невозможно. С уверенностью можно сказать только то, что такой путь всегда существует.

Формат входного файла

Входной файл в первой строке содержит два натуральных числа N и M ($10 \leq N, M \leq 5000$) — размеры игрового поля. Во второй строке два натуральных числа R ($1 \leq R \leq 10$) — радиус шайбы и K ($0 \leq K \leq 100$) — количество перегородок. Следующие K строк, содержат целые числа, описывающие проемы в перегородках: первое число T ($1 \leq T \leq 40$) — количество проемов в перегородке и далее $T + 1$ чисел, разделенных пробелами, первое из которых — абсцисса перегородки, а следующие T чисел — ординаты верхних концов проемов, начиная с самого верхнего в порядке убывания. Абсциссы перегородок принадлежат отрезку $[R, M - R]$ и перечислены в порядке возрастания. Расстояние между любыми двумя перегородками не менее $2R$. Начало координат совпадает с левым нижним углом поля.

Формат выходного файла

Выходной файл содержит одно вещественное число с точностью не менее 10^{-6} — длину кратчайшего пути, пройденного шайбой.

Примеры

stdin	stdout
10 10 1 1 1 3 6	8.000000

Задача F. Идеальные наборы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

*Из всех искусств важнейшим является искусство идеального взвешивания.
„История Новогиреевки“, том 1.*

На базарах города Новогиреевка все продавцы взвешивают товары с помощью наборов гирь, каждая из которых весит целое число граммов, а суммарный вес всех гирь составляет N граммов. Такой набор называется *идеальным*, если любой груз весом в целое число граммов от 1 до N может быть уравновешен некоторым количеством гирь данного набора, и притом единственным образом. Груз всегда кладется на левую чашу весов, гири — на правую. Два способа взвешивания, отличающиеся лишь заменой некоторых гирь на другие того же веса, считаются одинаковыми. Для $N = 5$ идеальных наборов три: $(1, 1, 1, 1, 1)$, $(1, 1, 3)$ и $(1, 2, 2)$.

Каждый продавец на рынке предпочитает иметь собственный набор гирь. Найдите для каждого i -го продавца все идеальные наборы, имеющие наименьшее количество гирь общим весом в N_i граммов.

Формат входного файла

В первой строке указано количество продавцов на базаре T ($1 \leq T \leq 100$). Во второй строке записаны через пробел T натуральных чисел N_1, N_2, \dots, N_T , где N_i — общий вес всех гирь для i -го продавца ($1 \leq N_i \leq 10^5$).

Формат выходного файла

Для каждого i -го продавца выведите в первой строке два числа: M — количество идеальных наборов, имеющих наименьшее количество гирь, и K — количество гирь в каждом таком наборе, затем в следующих M строках — идеальные наборы из K гирь общим весом N_i граммов, перечисленных в порядке неубывания массы. Все наборы выведите в лексикографическом порядке.

Примеры

stdin	stdout
2	1 1
1 5	1
	2 3
	1 1 3
	1 2 2

Задача G. Розы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сеня любит розы и выращивает их у себя в огороде. Ещё Сеня любит деньги, поэтому продаёт свои розы на ближайшем рынке. Перед походом на рынок Сеня пересчитал лепестки у каждой розы и решил, что розы с K лепестками он будет продавать за R рублей. Если лепестков в розе меньше K , то отсутствие каждого из лепестков уменьшит цену на 1 рубль. Если же лепестков больше K , то такую розу Сеня будет продавать в 2 раза дороже — за $2R$ рублей. К концу дня Сеня продал все свои розы и хочет подсчитать свою выручку.

Формат входного файла

Первая строка содержит 3 числа K ($1 \leq K \leq 100$), R ($K \leq R \leq 1000$) и N ($1 \leq N \leq 100$). Вторая строка — N чисел, не превосходящих 100, — количество лепестков у каждой из N роз.

Формат выходного файла

Одно число — выручка Сени.

Примеры

<code>stdin</code>	<code>stdout</code>
10 20 3 10 5 15	75

Задача Н. Граф из треугольников

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано n вершин, $n = 3^k$. Разрешена следующая операция: выбрать три вершины x , y и z , и провести в графе три ребра: между x и y , x и z , y и z . Необходимо построить полный граф, то есть провести все рёбра, причём ни одно ребро нельзя проводить более одного раза. Граф неориентированный.

Формат входного файла

Одно число n ($n < 1000, n = 3^k, k > 0$).

Формат выходного файла

В первой строке нужно вывести p — количество треугольников. В последующих p строках вывести по три числа — номера вершин очередного треугольника. Вершины нумеруются с 1 до n .

Примеры

stdin	stdout
3	1 1 2 3

Задача I. Цена вопроса

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Дан неориентированный связный граф без петель и кратных рёбер из n вершин и m рёбер. Каждое из m рёбер имеет вес от 0 до 10^9 . Изначально каждая вершина определяет компоненту, состоящую из одной вершины — самой себя. Разрешена операция объединения двух компонент ребром, но каждая такая операция имеет свою стоимость, которая определяется следующим образом.

Пусть множества вершин A и B уже соединены в компоненты. Тогда цена объединения A и B равна максимуму из всех весов рёбер, которые существуют между A и B . То есть цена равна максимуму из весов всех рёбер u, v таких, что $u \in A, v \in B$.

Найдите минимальную стоимость соединения всех вершин в одну компоненту.

Формат входного файла

В первой строке записаны целые числа n и m ($1 \leq n \leq 5000, 0 \leq m \leq 300000$). В последующих m строках описываются рёбра тройками целых чисел $u \ v \ c$, где c — цена ребра, а u и v — соединяемые вершины.

Формат выходного файла

В первой строке необходимо вывести два числа: стоимость соединения и количество рёбер. В следующих строках — сами рёбра.

Примеры

<code>stdin</code>	<code>stdout</code>
--------------------	---------------------

Задача J. Феноменальная память

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Мистер Джон Смит обладает феноменальной памятью. Он много читает и хорошо помнит прочитанные тексты. Недавно Смит прочел новую книгу, состоящую из слов. Из M слов. Внезапно, Джон осознал, что текст ему хорошо знаком. В сознании Смита всплыла книга, прочитанная им в прошлом году, которая состояла из слов. Из N слов. Осталось выяснить, насколько похожи эти две книги.

Формат входного файла

В первой строке целое число N ($1 \leq N \leq 10^5$). Далее N строк S_i , состоящих из строчных латинских букв. Суммарная длина строк не превышает 10^5 . Далее целое число M ($1 \leq M \leq 10^6$). Потом M строк T_i , состоящих из строчных латинских букв. Суммарная длина строк T_i не превышает 10^6 .

Формат выходного файла

Для каждого символа каждой строки T_i вывести длину максимального префикса одной из строк S_j , совпадающего с подстрокой строки T_i , начинающейся с этого символа.

Примеры

stdin	stdout
5 acbd caba abar book bat 2 abacaba abarbabooca	3 2 2 4 3 2 1 4 2 1 0 2 2 3 0 0 2 1
2 icl xiii 2 xiiiicl icicx	4 1 1 1 3 0 0 2 0 2 0 1

Задача К. Шифр

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задача в проблемсете турнира *ICL'13* оказалась чрезвычайно сложной. Долгие часы конкурса не привели ни одну команду к верному решению, но одному из участников пришел в голову хитроумный план. Ему стало известно, что решение задачи скрыто в сейфе, который открывается при помощи некоторой последовательности чисел.

Про последовательность известно следующее:

1. Все элементы последовательности — делители некоторого числа N ;
2. Ни один из этих элементов не делится на другой;
3. Последовательность имеет максимально возможную длину;
4. Из всех последовательностей, удовлетворяющих предыдущим трем пунктам, искомая — лексикографически наименьшая.

Откройте сейф.

Формат входного файла

Единственное целое число N ($1 \leq N \leq 2^{63} - 1$). Гарантируется, что число N не имеет простых делителей, превосходящих 100.

Формат выходного файла

В первой строке одно число — количество чисел в последовательности. Во второй строке сами числа, разделенные пробелом.

Примеры

<code>stdin</code>	<code>stdout</code>
12	2 2 3