Задача A. Деревянная доска

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

Петя и Вася очень любят решать интересные задачи. У них есть одна на двоих прямоугольная доска. Доска — это прямоугольник со сторонами, параллельными осям координат, и вершинами $(0, 0)$ и (W, H) в левом нижнем и правом верхнем углах соответственно. Она предназначена для того, чтобы кто-нибудь из них вешал на нее листочки с условиями задач, которые они не решили. Листочки имеют форму прямоугольников (прямоугольники могут вырождаться в точку), причем Петя и Вася вешают их на специальные гвоздики, которые не позволяют поворачиваться листочку вокруг гвоздика. Петя и Вася — весьма аккуратные мальчики, поэтому вешают листочек так, что его стороны были параллельны осям координат. Гвоздик держит листочек, если точка, которая соответствует гвоздику, находится внутри или на границе листочка.

В каждый момент времени один из друзей может:

- Повесить листочек на гвоздик, при этом можно его вешать поверх других листочков. Гвоздь прокалывает все листочки, внутри или на границе которых он находится.
- Снять листочек с номером k с доски, при этом все гвоздики, которые его прокалывают, падают. Тем самым, помимо листочка с номером k, могут упасть и другие листочки. Листочки нумеруются, начиная с единицы, в том порядке, в котором их вешали на доску.
- Узнать, сколько гвоздиков прокалывают листочек с номером k. Если листочек упал, то это значение равно 0.
- Узнать, сколько всего листочков осталось.

Помогите Пете и Васе ответить на их вопросы.

Формат входного файла

Первая строка входного файла содержит два целых числа W и H ($5 \le W, H \le 10^8$) — ширина и высота доски.

Во второй строке находится число N ($1 \le N \le 10000$) — число запросов.

В следующих N строках находятся описания запросов. Запрос может иметь один из четырех типов:

- $1\ x_1\ y_1\ x_2\ y_2\ x\ y$ — Повесить листочек так, чтобы его левый нижний угол имел координаты (x_1, y_1), правый верхний — (x_2, y_2) и прикрепить его гвоздиком в точке с координатами (x, y). $(0 \le x_1 \le x \le x_2 \le W)$, $(0 \le y_1 \le y \le y_2 \le H)$. В одной точке может быть несколько гвоздиков.
- $2\ k$ — Снять листочек с номером k. Гарантируется, что кто-нибудь из мальчиков до этого момента вешал на доску листочек с номером k.
- $3\ k$ — Узнать, сколько гвоздиков на листочке с номером k. Гарантируется, что кто-нибудь из мальчиков до этого момента вешал на доску листочек с номером k.
- 4 — Узнать, сколько всего листочков осталось на доске.

Все числа во входном файле целые.

Формат выходного файла

В выходной файл выведите ответы на запросы с номерами типов 3 и 4, по одному целому числу на строке.

Примеры

billboard.in	billboard.out
10 10 8 1 2 2 8 8 3 3 4 1 0 0 5 5 5 5 3 1 1 5 5 10 10 5 5 2 2 4 3 3	1 2 1 0
100 100 6 1 5 5 5 5 5 5 1 5 5 5 5 5 5 4 3 1 2 2 4	2 2 1

Задача B. Монеты и гнезда

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

Петя долго-долго сидел у окна и наблюдал, как n ворон, каждая в свое гнездо, приносят монеты. Петя заметил, что i-ая ворона принесла в свое гнездо a_i монет. Опытным глазом Петя подметил, что если в i-ом гнезде окажется b_i монет, то гнездо со всем своим содержимым упадет на землю, и все монеты достанутся Пете.

У Пети есть m монет. Петя очень метко кидает монеты в гнезда. Помогите Пете узнать, какое максимальное число монет он может получить.

Формат входного файла

В первой строке входного файла находятся два целых числа n и m ($1 \le n \le 1000$, $0 \le m \le 1000$) — число ворон и монет у Пети соответственно. Во второй строке находятся n чисел a_i ($0 \le a_i \le 1000$). В третьей строке находятся n чисел b_i ($a_i < b_i \le 1000$).

Формат выходного файла

В выходной файл выведите одно число — максимальное число монет, которые Петя может получить.

Примеры

coins.in	coins.out
2 3 1 2 4 6	6
3 3 1 2 3 4 8 16	4

Задача C. Эй! Это МОЯ рыба!

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

Одна из популярных настольных игр «Эй! Это МОЯ рыба!» состоит в перемещении пингвинов и сбору рыбы на поле.

Мы рассмотрим упрощенную версию этой игры. Для этой игры используется n карточек с изображениями нескольких рыб и фишки-пингвины.

В начале игры карточки выкладываются на столе в ряд в некотором порядке.

Каждому игроку выдается по два пингвина. Затем первый игрок размещает одного из своих пингвинов на незанятой льдине. После этого второй игрок делает то же самое. После этого они аналогично размещают своих вторых пингвинов.

Как только все пингвины находятся на льдинах, лов рыбы начинается! Игроки ходят по очереди. Ход состоит из перемещения одного из пингвинов текущего игрока, после этого игрок берет льдину — карточку, на которой его пингвин стоял в начале его хода, и кладет ее перед собой.

Пингвин может двигаться по льдинам любом направлении и остановиться на любой незанятой льдине. При этом запрещено перепрыгивать через других пингвинов и через проруби, оставшиеся после удаления льдин. Если у пингвина с обеих сторон препятствие, то он не может двигаться.

Если игрок не может переместить ни одного из его пингвинов, он забирает всех своих пингвинов и льдины, на которых эти пингвины стояли. После этого второй игрок продолжает делать ходы, пока он может перемещать своих пингвинов.

В конце игры каждый игрок считает количество рыб на всех собранных им карточках. Цель игрока максимизировать разность своего количества рыб и количества рыб противника. Какую максимальную разность может получить первый игрок, при оптимальной игре второго игрока.

Формат входного файла

В первой строке входного файла число n ($4 \le n \le 10$) — количество карточек. Во второй строке файла находится n чисел a_i — количество рыб на карточках в порядке в котором они выложены на столе. $1 \le a_i \le 10^6$.

Формат выходного файла

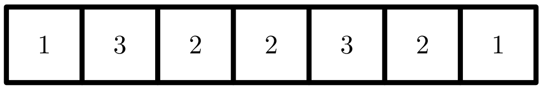
В выходной файл выведите одно число: какую максимальную разность может получить первый игрок, при оптимальной игре второго игрока.

Примеры

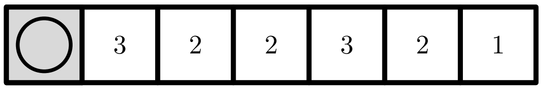
fish.in	fish.out
7 1 3 2 2 3 2 1	0

Пример игры для теста из условия

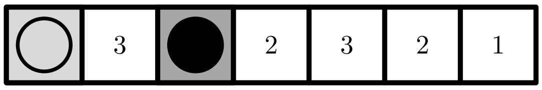
В начале игры на поле нет ни одного пингвина:



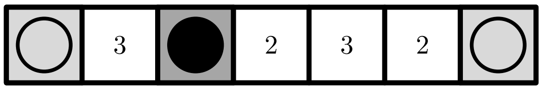
Первый игрок выставляет первого пингвина:



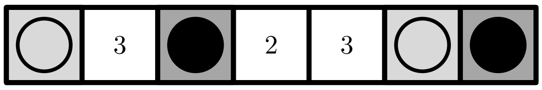
Второй игрок выставляет первого пингвина:



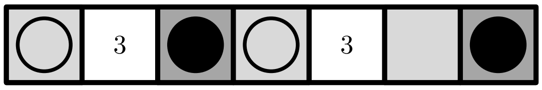
Первый игрок выставляет своего второго пингвина:



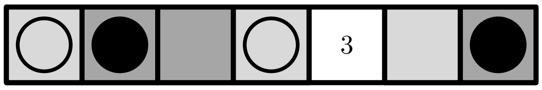
Второй игрок выставляет своего второго пингвина:



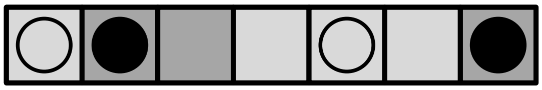
Ход первого игрока:



Ход второго игрока:



Ход первого игрока:



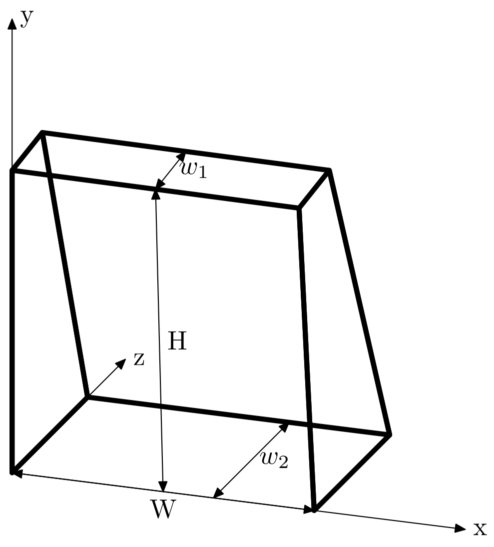
Задача D. Хоккейные ворота

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

Министр по делам семьи, молодежи и спорта Флатландии решил сделать игру хоккей «национальным» видом спорта. Для этого он приказал в каждом населенном пункте построить хоккейные площадки, в каждом городе — ледовый дворец, а также закупить все необходимое оборудование (например, ворота, клюшки, шайбы).

После исследования рынка спортивных товаров обнаружилось, что хоккейные ворота покупать очень дорого, поэтому их решили делать самостоятельно. Но вот незадача: никто не знает площадь сетки, которую нужно натянуть на одни ворота.

Ворота выглядят следующим образом:



Высота ворот — H метров, ширина — W метров, малая глубина — w_1 метров, большая глубина — w_2 метров. Сетку натягивают на заднюю часть ворот, бока и верхнюю часть ворот. Заметим, что для большей экономии сетку натягивают так, чтобы ее площадь была минимальной.

Уже скоро начнутся первые Флатландские игры, а Флатландские математики все еще не могут оценить затраты на ворота. Министр узнал о ваших успехах в области программирования и попросил написать программу, решающую его задачу.

Формат входного файла

В единственной строке находятся четыре вещественных числа H, W, w_1, w_2 ($0 < H, W \leq 2$, $0 \leq w_1 \leq w_2 \leq 2$).

Формат выходного файла

В выходной файл выведите единственное число — общую площадь сетки, необходимую для одних хоккейных ворот, с точностью до 10^{-5}.

Примеры

hockey.in	hockey.out
`1.22 1.83 1 1.5`	`7.29283`
`2 1.5 1 1.5`	`9.59233`

Задача E. Игра

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

Два игрока играют в новую настольную игру. На поле для игры есть города и дороги между ними, причем дороги между разными городами могут иметь различную длину. При этом, если длина дороги между двумя городами равна x, то на этой дороге в ряд находятся $(x-1)$ деревень, можно перейти от города к крайней деревне на дороге, от этой деревне к соседней, и т. д.

У каждого их игроков есть по фишке, исходно фишки игроков расположены в некоторых различных городах. Игроки ходят по очереди: за один ход можно передвинуть фишку в город или деревню, которые соединены с текущей вершиной напрямую, либо можно оставить фишку на месте. Начинает первый игрок.

Однако есть одно ограничение: если один из игроков находится в одной из деревень на дороге между двумя городами, то другой не может переходить ни в одну деревню, находящуюся на этой дороге.

Цель каждого из игроков заключается в том, чтобы добраться в столицу первым.

Формат входного файла

В первой строке входного файла находятся два числа n — число городов и m — число дорог $(1 \le n, m \le 10^5)$.

В следующих m строках находятся по три числа a, b и c, означающих, что между городами a и b есть дорога длиной c, содержащая $c-1$ деревню ($a \ne b$, $1 \le c \le 10^9$).

В последней строке находится три числа s_1, s_2 — номера городов в которых изначально находятся фишки игроков ($s_1 \ne s_2$), и t — номер города, являющегося столицей.

Гарантируется, что по дорогам из любого города можно добраться в любой другой.

Формат выходного файла

В выходной файл выведите «`First`», если первым в столицу доберется первый игрок, и «`Second`», если второй.

Примеры

game.in	game.out
4 3 1 3 2 3 2 3 4 3 1 1 4 2	Second

Задача F. Герою море по колено

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

В одной волшебной стране все еще живы драконы, держащие плену благородных принцесс, а также отважные рыцари, драконов побеждающие и принцесс спасающие. Про нелегкую судьбу тружеников меча и будет эта задача.

Как-то раз держал путь Витольд III из родного замка Златоперый Орел к ущелью Рокового Кузнечика, в котором обосновался очередной дракон. Между замком и ущельем раскинулась ровная как стол долина. Единственное препятствие на пути героя — странное болото, имеющее форму треугольника. Как известно, герою все нипочем, и болото ему не преграда. Тем не менее, скорость передвижения боевого коня по болоту заметно меньше, чем по земле.

Ваша задача — помочь дракону подготовиться к приезду рыцаря. Недавно стало известно, что Витольд устроил привал у одной из вершин треугольного болота. Вам необходимо узнать, какое минимальное время потребуется рыцарю, чтобы добраться от замка до логова, если он сей же час отправится в путь.

Формат входного файла

Первая строка входного файла содержит шесть целых чисел, разделенных пробелами: x_1, y_1, x_2, y_2, x_3 и y_3 — координаты углов треугольного болота ($|x_1|, |y_1|, |x_2|, |y_2|, |x_3|, |y_3| \le 1000$).

Вторая строка входного файла содержит два целых числа: x_d и y_d — координаты логова дракона ($|x_d|, |y_d| \le 1000$).

Последняя строка входного файла содержит два натуральных числа: v_1 и v_2 — скорость передвижения по суше и по болоту соответственно ($1 \le v_2 < v_1 \le 1000$).

Витольд находится в первой вершине болота, то есть в точке с координатам (x_1, y_1).

Формат выходного файла

В выходной файл выведите одно число — минимальное время, которое потребуется рыцарю, чтобы добраться от замка до ущелья. Ответ требуется вывести с точностью не менее 10^{-6}.

Примеры

move.in	move.out
0 0 2 -1 2 1 4 0 2 1	2.236068
0 0 2 -1 2 1 1 0 2 1	0.834512
0 0 10 0 10 10 9 5 5 1	3.979796

Задача G. Рекламный плакат

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

Однажды в одном далеком-далеком городе БайтСити компания МакроХард решила разместить по всему городу свои рекламные плакаты. Для этих плакатов было заготовлено несколько рекламных лозунгов.

Каждый лозунг состоит из нескольких слов, порядок которых, разумеется, менять нельзя. Каждое следующее слово можно расположить на плакате или в новой строке горизонтально, или, начиная со следующей строки, вертикально по одной букве слова в одной строке. При этом нельзя два последовательных слова написать вертикально, так как в этом случае их нельзя будет различить.

Известно, что рекламный плакат можно сделать сколь угодно высоким, но, к сожалению, ширина плаката ограничена. Поэтому компания МакроХард хочет разместить лозунг на плакате таким образом, чтобы минимизировать максимальную ширину. Каждое слово печатается одним и тем же моноширинным шрифтом, потому достаточно минимизировать длину самого длинного слова, записанного горизонтально.

В данной задаче Вам дан один рекламный лозунг. Необходимо найти размещение, минимизирующее ширину плаката.

Формат входного файла

В первой строке входного файла задано одно число n ($1 \le n \le 100$) — число слов в рекламном лозунге. В следующих n строках записан сам лозунг — по одному слову в строке.

Длина каждого слова не менее двух и не более 100 символов. Слова состоят только из строчных латинских букв.

Формат выходного файла

В выходной файл выведите единственное число — минимальную ширину плаката.

Примеры

poster.in	poster.out
5 THIS IS AN EASY EXAMPLE	4
9 AND THIS IS SECOND SAMPLE FOR THIS EASY PROBLEM	6

Задача H. Ролевая игра

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

Когда у Жени бывает свободное время, он любит тратить его на свою любимую ролевую игру. Конечно же, у него есть мечта — достичь 255 уровня, то есть набрать необходимое количество очков опыта. К сожалению, у него не так много свободного времени, потому он хочет использовать его максимально эффективно и просит Вас помочь ему в этом.

Набирать очки опыта в игре можно посредством выполнения некоторых заданий. Для каждого задания известно время его выполнения, сколько очков оно приносит и список заданий, которые необходимо выполнить до того, как это задание станет доступно.

Помогите Жене выяснить, какое максимальное количество опыта, он сможет набрать за свободное время.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \le n \le 16, 1 \le m \le 1000$) — количество заданий и количество свободного времени у Жени в минутах. В следующих n строках следуют описания заданий по одному в каждой строке в следующем формате: сначала два целых числа t_i, p_i и k_i ($1 \le t_i, p_i \le 1000, 0 \le k_i < n$) — время в минутах, которое занимает выполнение i-го задания, сколько опыта получит персонаж Жени при его выполнении, и количество заданий, которые необходимо выполнить до i-го, соответственно. Далее следуют k_i различных чисел от 1 до n — номера заданий, которые необходимо выполнить до i-го.

Формат выходного файла

В первой строке выведите единственное целое число — максимальное количество опыта, которое Женя может набрать за свободное время. В следующей строке выведите номера заданий, которые необходимо выполнить для получения такого количества опыта, в том порядке, в котором их необходимо выполнять.

Примеры

rpg.in	rpg.out
2 10 5 3 1 2 5 2 0	5 2 1
3 9 4 1 0 5 2 1 1 6 3 1 1	3 1 2

Задача I. Стадион

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

Небольшое островное государство Кусмания ничем не примечательно. Сколько кусманцы ни старались, они не нашли у себя ни археологических ценностей, ни полезных ископаемых, ни потенциальных мест для организации курортного бизнеса. Этот факт не дает покоя кусманскому правительству, ведь должен же их народ быть хоть чем-то знаменит! В такой печальной ситуации остается одно — спорт.

Для того, чтобы добиться успехов в спорте, было решено построить стадион. Чем больше стадион, тем больше спортсменов смогут там тренироваться. Проблема только в том, что острова, относящиеся к территории Кусмании, невелики, поэтому размеры стадиона ограничены размерами острова.

С высоты птичьего полета стадион имеет форму идеального круга, а острова — форму выпуклых многоугольников. Требуется написать программу, которая по заданной форме острова находила бы максимальный радиус стадиона, который помещается на этом острове.

Формат входного файла

Первая строка входного файла содержит натуральное число n — число вершин многоугольника ($3 \le n \le 100$). Каждая из следующих n строк содержит по два целых числа x_i и y_i — координаты i-й вершины в порядке обхода ($|x_i|, |y_i| \le 10000$). Гарантируется, что многоугольник выпуклый.

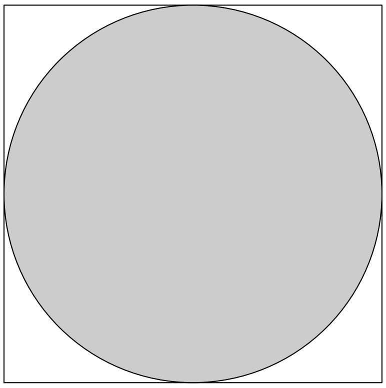
Формат выходного файла

В выходной файл выведите одно вещественное число — максимальный радиус стадиона, который можно построить на этом острове. Ответ требуется вывести с точностью не менее 10^{-6}.

Примеры

stadium.in	stadium.out
4 0 0 1 0 1 1 0 1	0.5

Пояснение к примеру:



Задача J. Сколько звезд на небе?

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

Британские ученые решили посчитать, сколько звезд на небе. Для этого они построили супермегателескоп, в который, если никто не мешает, можно разглядеть муху на поверхности Альфа Центавры. Чтобы обрабатывать данные, поступающие с этого телескопа, они построили супермегакластер, способный просчитать движение всех звезд Млечного Пути на сорок восемь миллионов лет вперед. Для решения проблемы энергоснабжения этого кластера британские ученые обратились к своим швейцарским друзьям, и те поделились с ними энергией Большого Адронного Коллайдера. И вот процесс начался.

Итак, на небе N звезд. Так как для упрощения модели небо решили аппроксимировать плоскостью, то у каждой звезды есть координаты X_i, Y_i. Координаты звезд вычислены с поражающей воображение точностью, поэтому можно считать, что ни у каких двух звезд координаты не совпадают.

Теперь британские ученые хотят собрать статистику. Для этого они сформулировали M запросов, каждый из которых звучит так: «Сколько звезд находится внутри или на границе области, заданной следующими неравенствами: $X_j^{min} \le x \le X_j^{max}$, $Y_j^{min} \le y \le Y_j^{max}$?»

К сожалению, из-за неожиданного извержения вулкана британские ученые не смогли собраться на очередное заседание, чтобы вычислить результаты запросов. Поэтому они просят Вас помочь им.

Формат входного файла

В первой строке входного файла даны два целых числа N ($1 \le N \le 100000$) и M ($1 \le M \le 50000$) — число звезд на небе и число запросов соответственно.

Далее в N строках заданы координаты звезд — пары целых чисел (X_i, Y_i) ($|X_i|, |Y_i| \le 10^9$). Никакие две звезды не совпадают.

Далее в M строках заданы запросы — четверки целых чисел $(X_j^{min}, X_j^{max}, Y_j^{min}, Y_j^{max})$. Все величины в описании запросов не превосходят 10^9 по абсолютному значению. Гарантируется, что $X_j^{min} \le X_j^{max}$, $Y_j^{min} \le Y_j^{max}$.

Формат выходного файла

Для каждого запроса в отдельной строке выведите ответ на этот запрос.

Примеры

stars.in	stars.out
1 1 30 239 13 42 11 100500	1
4 2 0 0 2 0 0 2 2 2 -1 3 -1 3 1 5 1 5	4 1