

Задача А. Ответ

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

Многие не поверили, что ответ на главный вопрос жизни, вселенной и всего такого — 42. Большой Компьютер огорчился этим фактом, поэтому он попытался найти ответ ещё раз. Для этого он постороил другой компьютер, который выдал ему набор из n натуральных чисел. Никакие два из этих чисел не совпадали. Большой Компьютер решил, что если в этом списке найдутся три различных числа такие, что записав их подряд мы получим число-палиндром, то есть число, читающееся одинаково как слева направо, так и справа налево, то ответом будет сам этот палиндром. При этом лидирующие нули в записи числа-палиндрома не допустимы.

Внезапно по телевизору стали показывать очень интересный для Компьютера мультик, и поэтому он поручил вам найти ответ на эту задачу.

Формат входного файла

В первой строке входного файла задано единственное натуральное число n ($1 \leq n \leq 900$) — количество сгенерированных чисел. Во второй строке заданы сами числа. Все числа попарно различные, целые, положительные и не превышают 10^9 .

Формат выходного файла

В единственную строку выходного файла выведете через пробел три числа из которых состоит палиндром. Если ответов несколько — выведите любой. Если ответа не существует — выведете: «Sorry, but I don't know answer!».

Примеры

<code>answer.in</code>	<code>answer.out</code>
4 12 21 45 3	12 3 21
1 42	Sorry, but I don't know answer!

Задача В. Шифровка

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

Петя и Вася любят играть в шпионов. А какие игры в шпионов обходятся без секретных шифровок! Вот и Вася недавно придумал новый способ шифрования своих сообщений. Информацию о том, как он шифрует, он оставил в секрете и сообщил Пете только способ расшифровки, чтобы тот всегда мог им воспользоваться.

Для расшифровки Пете необходимо всего лишь найти лексикографически максимальную подстроку в зашифрованном сообщении, которое ему передает Вася. Именно эта подстрока и будет исходным текстом. Поскольку Петя справляется с этим поиском не так быстро, как ему хотелось бы, он просит Вас написать программу, которая поможет ему в этом!

Формат входного файла

Первая и единственная строка входного файла содержит непустое зашифрованное сообщение, состоящее только из строчных букв латинского алфавита. Сообщение имеет длину не более 1000 символов.

Формат выходного файла

В выходной файл выведите расшифрованное сообщение.

Примеры

<code>cipher.in</code>	<code>cipher.out</code>
abacaba	caba
babb	bb

Задача С. Совпадения случайны

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

Теперь в Берляндии новый национальный спорт — спортивное программирование! Для его популяризации и организации была создана Организация, целью которой стало проведение онлайн-соревнований. Однако получилось так, что сначала Организация провела регистрацию n участников, присвоила каждому некоторый рейтинг и провела m соревнований, а только потом приняла регламент проведения соревнований и правила участия в них.

Самым главным новшеством стало введение запрета использования нескольких различных аккаунтов одним человеком, и сейчас Организация пытается определить всех потенциальных нарушителей. Как показал многолетний опыт, участник A может быть нарушителем только в том случае, если есть участник B такой, что выполнены все следующие условия:

1. A и B никогда не участвовали в одном соревновании
2. рейтинг B после последнего соревнования меньше, чем рейтинг A
3. A и B писали соревнования с одного IP-адреса
4. B выступал на уровне A во всех соревнованиях, в которых принимал участие. Считается, что участник B выступил на уровне участника A в некотором соревновании, если для любого $i < k$ верно что $highest(i) \leq points(b)$ и $highest(k) > points(b)$, где $points(b)$ — результат участника B в этом соревновании, k — конечный рейтинг участника A , $highest(j)$ — максимальное число баллов, набранное в текущем соревновании участником, чей конечный рейтинг равен j . При этом считается, что участник, набравший в соревновании максимальное число баллов, выступил в нем на уровне участника с рейтингом десять. Если участник не участвовал ни в одном соревновании, его уровень считается равным нулю.

Вам как ведущему аналитику Организации было поручено по результатам соревнований определить всех участников, которые могут быть нарушителями.

Формат входного файла

В первой строке входного файла записано два целых числа: n и m ($1 \leq n, m \leq 200$) — количество зарегистрированных участников и количество соревнований. В следующих n строках записаны данные участников — в i -й строке записан логин s_i и текущий рейтинг r_i ($0 \leq r_i \leq 10$) участника с номером i , записанные через пробел. Логин участника может содержать только латинские буквы и может быть не длиннее 20 символов.

Далее находится m блоков, описывающих результаты соревнований. В первой строке каждого блока записано одно целое число k ($1 \leq k \leq n$) — количество участников, принимавших участие в этом соревновании. В следующих k строках записаны результаты каждого участника в следующем формате: логин участника s_j , целое число, показывающее количество набранных им в этом соревновании баллов, b_j ($0 \leq b_j \leq 2000$), целое число, описывающее изменение его рейтинга, d_j ($-10 \leq d_j \leq 10$) и IP-адрес, с которого этот участник писал констест, разделенные пробелами. При этом рейтинг участника не может опускаться ниже нуля и подниматься выше десяти. IP-адреса участников в течение серии соревнований не менялись. Каждый IP-адрес содержит только точки и цифры.

Формат выходного файла

В первой строке выходного файла выведите число l подозрительных участников. В следующих l строках выведите их логины в лексикографическом порядке.

Примеры

coincidence.in	coincidence.out
3 2 Petya 1 Vasya 2 Gena 3 2 Vasya 2 -2 10.0.0.1 Petya 10 2 10.0.0.2 2 Gena 100 0 10.0.0.1 Petya 10 0 10.0.0.2	1 Gena
2 2 Petya 10 Gena 3 1 Gena 1000 0 10.0.0.1 1 Petya 1000 0 10.0.0.1	1 Petya

Задача D. Новая игра

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

Все дети очень любят играть. Не являются исключением и главные герои этой задачи — Яша и Антон. При каждой новой встрече они придумывают новую игру и с азартом выясняют кто же лучший. Так вот и сегодня, когда они встретились во дворе, Яша предложил новую игру.

Яша записывает на листочке матрицу чисел размером $n \times m$ и помещает в одну из ячеек матрицы фишку. После этого начинается игра. Ходы делаются игроками по очереди и заключаются в том, что если в ячейке где находится фишка записано положительное число x , то игрок, делающий ход, может переставить фишку либо на x клеток вправо, либо на x клеток вниз. А если в ячейке записано отрицательное число $-x$, то игрок может переставить фишку либо на x клеток вверх, либо на x клеток влево. Выходить за пределы матрицы нельзя. Если игрок не может сделать ход, то он проиграл. Антон начинает игру первым.

Формат входного файла

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 1000$) — размеры матрицы. Далее следуют n строк по m чисел в каждой — описание матрицы игры. Все числа в матрице по модулю не превосходят 1000. Последняя строка содержит два числа — первоначальные координаты фишки. Гарантируется, что фишка изначально стоит в поле. Первое число обозначает номер строки, второе — номер столбца.

Формат выходного файла

В выходной файл выведите либо имя победителя игры на данной матрице («Yasha», если победит Яша, или «Anton», если победит Антон), либо слово «draw», если никто из них так и не сможет выиграть.

Считайте, что мальчики всегда играют по оптимальной стратегии.

Примеры

<code>game.in</code>	<code>game.out</code>
2 2 1 2 2 1 1 1	Anton
3 3 1 3 -1 3 -1 -1 1 -3 1 2 3	Yasha
3 3 1 1 -1 2 -1 -2 1 -2 -1 3 3	draw

Задача Е. Марсианский друг

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

Десятиклассник Вася дружит с марсианином Хрулем. И сегодня Хруль попросил Васю помочь решить следующая задачу: необходимо найти на какое количество нулей заканчивается число $N!$. Вася сначала обрадовался, ведь он решал такую задачу в пятом классе на олимпиадном кружке по математике. Но вовремя вспомнил, что на Марсе k -ичная система счисления, и задумался.

Помогите Васе посчитать количество нулей в записи число $N!$ в k -ичной системе счисления.

Формат входного файла

Входной файл содержит два целых числа $N(1 \leq N \leq 10^{18})$ и $k(2 \leq k \leq 10^9)$.

Формат выходного файла

В выходной файл выведите одно целое число — ответ на задачу.

Примеры

<code>mars.in</code>	<code>mars.out</code>
20 7	2

Задача F. Деловые встречи

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

Алексей — успешный предприниматель и в течении одного дня у него бывает много встреч с разными деловыми партнерами. К сожалению, встречи бывают разные и не все приносят ему радость и после них настроение улучшается. Также, на многие встречи не стоит приходить в слишком плохом или хорошем настроении — результат таких встреч может быть не таким, какой хочется Алексею.

К счастью, недавно Алексей научился оценивать свое настроение с помощью целых чисел. После этого для каждой встречи он оценил при каком максимальном и минимальном настроении стоит на нее приходить, а также как изменится его настроение после этой встречи. Теперь он хочет распланировать порядок встреч так, чтобы в течении дня совершить максимальное число встреч.

Ваша задача — написать программу, которая по информации о всех встречах и настроении Алексея в начале дня находит порядок проведения встреч такой, что их количество при этом максимально.

Формат входного файла

Первая строка входного файла содержит два целых числа n и k ($1 \leq n \leq 20$, $|k| \leq 100$) — количество встреч и настроение Алексея в начале дня.

Следующие n строк содержат по три целых числа a_i , b_i и c_i ($|a_i|, |b_i|, |c_i| \leq 100$) — минимальное, максимальное настроение при котором встреча возможна и изменение настроения по окончании встречи, соответственно.

Формат выходного файла

В первой строке выходного файла выведите число m — максимально возможно число встреч. В следующей строке выведите m целых чисел — номера встреч в порядке их проведения. Встречи пронумерованы в порядке описания во входном файле.

Если ответов с максимальным числом встреч несколько, выведите любой.

Примеры

<code>meetings.in</code>	<code>meetings.out</code>
3 0 1 3 3 0 1 2 1 3 1	3 2 3 1
3 1 -10 -5 3 -5 5 -2 -3 2 1	2 3 2

Задача G. Сообщение

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

Петя и Вася продолжают играть в шпионов. На этот раз Петя прислал Васе закодированное сообщение, ключами для декодирования которого являются два простых числа p и q . Петя сообщил Васе, что эти числа связаны соотношением $q = 2p + 1$, и p - это k -е по величине простое число для которого существует простое q , удовлетворяющее выше написанному соотношению.

Помогите Васе прочитать сообщение — найдите такое p .

Формат входного файла

Единственная строка входного файла содержит целое число k ($1 \leq k \leq 30000$) — порядковый номер числа p .

Формат выходного файла

В выходной файл выведите единственное число p .

Примеры

<code>message.in</code>	<code>message.out</code>
1	2
4	11

Задача Н. Спасти наш мир!

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

На одной из планет, под названием TheBestWorld, живут много умных людей. Эта планета также известна своими двумя конкурирующими школами супергероев. Каждый малыш живущий на этой планете мечтает стать супергероем, но не у всех это получается.

Ученики разных школ всегда недолюбливали друг друга. И иногда дело даже доходило до того, как супергерои начали применять свои геройские силы для выяснения отношений.

В этом мире случилось несчастье. На TheBestWorld нападают темные силы. У генерала главнокомандующего армией супергероев есть в распоряжении $n + m$ супергероев, из них n — выпускники первой школы, m — второй. Главнокомандующий подсчитал количество супергероев k , которые смогут спасти лучшее, что у них есть, — планету.

На первый взгляд кажется, что главнокомандующему не понадобится помощь в выборе супергероев, но он вспомнил, что супергерои из разных школ раздражают друг друга. Но это происходит только тогда, когда два супергероя из разных школ находятся на расстоянии меньше d . Если такую пару героев позвать спасать мир, они не смогут продуктивно делать свою работу.

Проблема в том, что генерал забыл искомое d . Вам известны координаты каждого из супергероев. Генерал просит вас вычислить максимальное d , при котором он все еще может спасти мир. И выдать список супергероев, которые спасут мир при данном d .

Формат входного файла

В первой строке входного файла заданы три числа n , m и k ($1 \leq k$, $n, m \leq n + m \leq 200$) — количество супергероев, выпустившихся из первой и второй школы, и количество супергероев, которое необходимо, чтобы спасти мир, соответственно. Следующие $n + m$ строк описывают супергероев: в каждой строке заданы два целых числа — текущие координаты супергероя. Первые n строк содержат информацию о выпускниках первой школы, следующие m — о выпускниках второй школы.

Все координаты не превышает по модулю 10000.

Известно, что никакие два супергероя не стоят в одной точке.

Формат выходного файла

В первую строку выходного файла выведите одно вещественное число: максимальное значение d , которое позволяет спасти мир, либо -1, если при любом значении d мир можно спасти. Относительная или абсолютная погрешность вашего ответа не должна превышать 10^{-6} .

В следующих k строках выведите список номеров супергероев по одному числу в строке, которых можно взять при данном d , либо, если спасти мир можно при любом d , выведите список героев, которые позволяет спасти мир вне зависимости от d . Супергерои нумеруются числами от 1 до $n + m$ в том порядке, в котором они идут во входном файле.

Примеры

<code>savethebest.in</code>	<code>savethebest.out</code>
2 3 4	2.0
1 1	1
1 2	3
1 3	4
1 4	5
1 5	
1 1 2	9.0
1 1	1
1 10	2

Задача I. Суперзвезда

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

В альтернативной геометрии, изучаемой последователями доктора Иллитида, одной из ключевых фигур является n -угольная суперзвезда. Она используется для того, чтобы произносить заклинания, призывать Ктулху и делать много других крайне полезных вещей. Описание процесса построения фигуры приведено практически в любом справочнике для юных последователей, и состоит всего из трех пунктов:

- возьмите окружность радиусом 10 метров и поставьте на ней n точек так, что длины дуг между всеми парами соседних точек одинаковы
- пронумеруйте точки от 0 до $n - 1$, обходя окружность по часовой стрелке
- соедините отрезками те и только те точки, для которых верно равенство $(a + 2) \bmod n = b$, где a и b — номера соединяемых точек

После этого вы можете встать в центр окружности и призвать Ктулху, у которого будет ровно n щупалец. Все, что вам для этого понадобится — правильно назвать площадь нарисованной вами звезды. Напишите программу, решающую эту задачу за вас.

Формат входного файла

Входной файл содержит одно целое число n ($5 \leq n \leq 100$) — количество щупалец у Ктулху, которого вы хотите призвать.

Формат выходного файла

В выходной файл выведите одно целое число — ответ на задачу, выведенный с точностью не меньшей, чем 5 знаков после запятой.

Примеры

<code>superstar.in</code>	<code>superstar.out</code>
5	112.25699414489632

Задача J. Туристическое агентство

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

В одном большом городе очень много достопримечательностей и в него регулярно приезжают толпы туристов. Но просто приезжать в город неинтересно, и туристы любят осматривать достопримечательности. Все туристы — занятые люди и осматривать две достопримечательности одного и того же типа не собираются. Например, никто не пойдёт в два музея, ведь, чтобы сказать, что ты был в музее, достаточно сходить только в один.

Все n достопримечательностей расположены вдоль главной улицы города. Каждая достопримечательность имеет свой тип — музей, театр, памятник... Каждый приезжающий турист заказывает в турагентстве экскурсию. Экскурсия представляет из себя проезд по каким-то достопримечательностям, стоящим подряд. Так как туристы живут в разных частях города, то не всем им удобно добираться до главной улицы. Поэтому, для i -го туриста есть границы $[l_i; r_i]$ — отрезок, в который должны попасть начало и конец экскурсии, ведь ему надо не только приехать на неё, ну и уехать потом домой.

Каждый турист хочет осмотреть как можно больше достопримечательностей, но при этом он не будет смотреть более одной достопримечательности одного типа. Помогите турагентству подобрать каждому туристу подходящую ему экскурсию.

Формат входного файла

В первой строке входного файла задано число n ($1 \leq n \leq 100\,000$) — количество достопримечательностей в городе. Во второй строке заданы n целых чисел t_i ($1 \leq t_i \leq 10^9$) — типы достопримечательностей. В третьей строке задано число m ($1 \leq m \leq 100\,000$) — количество туристов. Далее, в m строках заданы описания туристов в формате $l_i r_i$ ($1 \leq l_i \leq r_i \leq n$), где l_i, r_i — отрезок, в который должны попасть начало и конец экскурсии для i -го туриста.

Формат выходного файла

В выходной файл для каждого туриста выведите количество осмотренных им достопримечательностей.

Примеры

<code>tourists.in</code>	<code>tourists.out</code>
5	2
1 1 2 2 1	1
5	2
1 5	1
1 2	1
2 3	
3 4	
1 1	