

Задача А. Покупка цветов

Имя входного файла: `flowers.in`
Имя выходного файла: `flowers.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

На День учителя Вася решил купить букет цветов. В магазине продаются ромашки по A рублей за штуку и гладиолусы по B рублей за штуку ($A < B$). У Васи есть C рублей. Он хочет составить букет из максимально возможного количества цветов, и при этом потратить как можно больше денег. Другими словами, из всех букетов с максимально возможным количеством цветов он хочет выбрать самый дорогой, но не дороже C рублей. Помогите ему вычислить стоимость такого букета.

Формат входных данных

Вводятся три целых числа A, B, C ($1 \leq A < B \leq 100, 0 \leq C \leq 1000$).

Формат выходных данных

Выведите одно число — стоимость самого дорогого букета из максимального количества цветов.

Примеры

<code>flowers.in</code>	<code>flowers.out</code>
2 3 11	11
3 5 10	9

Комментарии

В первом тесте искомый букет состоит из четырёх ромашек и одного гладиолуса. Во втором — только из трёх ромашек.

Задача В. Пицца

Имя входного файла: `pizza.in`
Имя выходного файла: `pizza.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Компания из M человек пришла в пиццерию. Посоветовавшись, они решили заказать одну большую пиццу с K начинками. Пицца представляет собой круг, поделённый на K равных секторов, в каждом из которых находится своя начинка. Пиццу подают ещё не разрезанной.

Друзья попросили официанта разрезать пиццу на M равных секторов, по одному куску на человека, так, чтобы как можно большему количеству людей достался кусок по крайней мере с двумя начинками.

Помогите официанту определить, какому именно количеству людей достанется больше одной начинки, если резать пиццу наиболее оптимально.

Формат входных данных

Вводятся два целых числа K, M ($1 \leq K \leq 100, 1 \leq M \leq 100$) — количество начинок в пицце и количество человек в компании соответственно.

Формат выходных данных

Выведите количество человек, которым достанется более одной начинки в наилучшем случае.

Примеры

	<code>pizza.in</code>	<code>pizza.out</code>
3 3		3
3 2		2

Комментарии

В первом тесте каждому достанется по две начинки, если резать как угодно, но не по границам секторов с начинками.

Во втором тесте не важно как резать: в любом случае обоим достанется по половине пиццы, в каждой из которых будет больше одной начинки.

Задача С. Решение задач

Имя входного файла: `solving.in`
Имя выходного файла: `solving.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

В этой задаче Вася готовится к олимпиаде. Учитель дал ему N ($1 \leq N \leq 100\,000$) задач для тренировки. Для каждой из этих задач известно, каким умением a_i нужно обладать для её решения. Это означает, что если текущее умение Васи больше либо равно заданного умения для задачи, то он может ее решить. Кроме того, после решения i -й задачи Васино умение увеличивается на число b_i .

Исходное умение Васи равно A . Решать данные учителем задачи он может в произвольном порядке. Какое максимальное количество задач он сможет решить, если выберет самый лучший порядок их решения?

Формат входных данных

Сначала вводятся два целых числа N, A ($1 \leq N \leq 100\,000, 0 \leq A \leq 10^9$) — количество задач и исходное умение. Далее идут N пар целых чисел a_i, b_i ($1 \leq a_i \leq 10^9, 1 \leq b_i \leq 10^9$) — соответственно сколько умения нужно для решения i -й задачи и сколько умения прибавится после её решения.

Формат выходных данных

Выведите одно число — максимальное количество задач, которое Вася сможет решить.

Примеры

<code>solving.in</code>	<code>solving.out</code>
3 2 3 1 2 1 1 1	3
4 1 1 10 21 5 1 10 100 100	3

Комментарии

В первом тесте Вася сможет решить все задачи, выбрав, например, порядок 2, 1, 3. Во втором тесте ему необходимо сначала разобраться с 1 и 3 задачами, после чего он осилит 2.

Задача D. Стильная одежда

Имя входного файла: `style.in`
Имя выходного файла: `style.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Глеб обожает шопинг. Как-то раз он загорелся идеей подобрать себе кепку, майку, штаны и ботинки так, чтобы выглядеть в них максимально стильно. В понимании Глеба стильность одежды тем больше, чем меньше разница в цвете элементов его одежды.

В наличии имеется N_1 кепок, N_2 маек, N_3 штанов и N_4 пар ботинок ($1 \leq N_i \leq 100\,000$). Про каждый элемент одежды известен его цвет (целое число от 1 до 100 000). Комплект одежды — это одна кепка, майка, штаны и одна пара ботинок. Каждый комплект характеризуется максимальной разницей между любыми двумя его элементами. Помогите Глебу выбрать максимально стильный комплект, то есть комплект с минимальной разницей цветов.

Формат входных данных

Для каждого типа одежды i ($i = 1, 2, 3, 4$) сначала вводится количество N_i элементов одежды этого типа, далее в следующей строке — последовательность из N_i целых чисел, описывающих цвета элементов. Все четыре типа подаются на вход последовательно, начиная с кепок и заканчивая ботинками. Все вводимые числа целые, положительные и не превосходят 100 000.

Формат выходных данных

Выведите четыре целых числа — цвета соответственно для кепки, майки, штанов и ботинок, которые должен выбрать Глеб из имеющихся для того, чтобы выглядеть наиболее стильно. Если ответов несколько, выведите любой.

Примеры

style.in	style.out
3 1 2 3 2 1 3 2 3 4 2 2 3	3 3 3 3
1 5 4 3 6 7 10 4 18 3 9 11 1 20	5 6 9 20

Задача Е. Подтасовка результатов

Имя входного файла: `juggling.in`
Имя выходного файла: `juggling.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

В городе Н. олимпиада по информатике состоит из двух туров, каждый из которых оценивается из 400 баллов. Для удобства все её участники занумерованы числами от 1 до N .

Сразу после проведения олимпиады курьер принёс жюри пренеприятнейшее известие: «сверху» пришло указание о том, что некто Вася, выступавший в олимпиаде под номером 1, должен по итогам олимпиады занять место A , то есть ровно $A - 1$ участников должны набрать по сумме двух туров больше баллов, чем Вася. При этом места, занятые школьниками в каждом из туров в отдельности, уже опубликованы, и их менять нельзя. Для каждого тура дан список номеров участников в порядке занятого места — перестановка чисел от 1 до N . Теперь задача жюри заключается в том, чтобы расставить целые баллы от 1 до 400 каждому участнику в первом и втором турах таким образом, чтобы в итоговой таблице Вася занял место A , а места участников в каждом из туров не изменились.

Никакие два участника не должны получить в одном туре одинаковые баллы. Одинаковые баллы в итоговой таблице возможны.

Ваша задача — проделать за жюри такую работу или определить, что это невозможно.

Формат входных данных

В первой строке вводятся два целых числа N , A ($1 \leq N \leq 200$, $1 \leq A \leq N$) — соответственно количество участников олимпиады и требуемое Васино место. Во второй строке перечислены номера участников в порядке занятых мест в первом туре (от первого места до N -го). В третьей строке в таком же формате следует описание второго тура. Номера участников во второй и третьей строках разделены пробелами.

Формат выходных данных

В случае, если невозможно расставить баллы требуемым образом, выведите единственное слово **Impossible**. Иначе в первой строке выведите **Possible**, во второй строке выведите N целых чисел от 1 до 400, соответствующих расстановке баллов участникам первого тура, где i -ое число — балл в первом туре участника, занявшего на нём i -е место, в третьей аналогично выведите N целых чисел, соответствующих расстановке баллов во втором туре. Числа в строках разделяйте пробелами.

Никакие два участника не должны получить одинаковые баллы в одном и том же туре. Если существует несколько способов расставить баллы требуемым образом, выведите любой.

Примеры

<code>juggling.in</code>	<code>juggling.out</code>
3 1 2 1 3 3 1 2	Possible 3 2 1 3 2 1
3 1 2 3 1 3 1 2	Impossible
5 3 2 3 4 5 1 2 3 1 4 5	Possible 5 4 3 2 1 400 399 398 2 1
5 3 1 2 3 5 4 5 4 1 3 2	Possible 5 4 3 2 1 400 399 3 2 1

Задача F. Прямоугольные треугольники

Имя входного файла: `trian.in`
Имя выходного файла: `trian.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Вася заасфальтировал один прямоугольный треугольник, а Петя забетонировал другой прямоугольный треугольник. Катеты каждого из треугольников параллельны осям координат. Необходимо определить, забетонировал ли Петя хотя бы одну заасфальтированную точку.

Формат входных данных

Вам даны 8 целых чисел: $x_1, y_1, a_1, b_1, x_2, y_2, a_2, b_2$, где (x_1, y_1) - координаты прямого угла первого треугольника, а остальные две вершины имеют координаты $(x_1 + a_1, y_1)$ и $(x_1, y_1 + b_1)$. Аналогично, (x_2, y_2) - координаты прямого угла второго треугольника, а остальные две вершины имеют координаты $(x_2 + a_2, y_2)$ и $(x_2, y_2 + b_2)$. Каждое число по модулю не превосходит 10^9 и может быть равно нулю.

Формат выходных данных

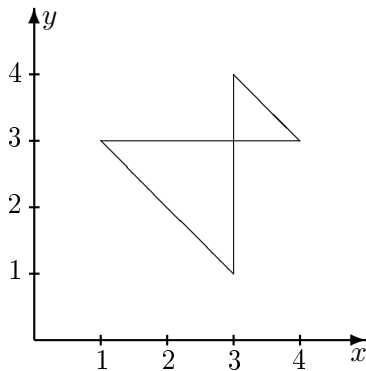
Выведите YES, если Петя забетонировал хотя бы одну заасфальтированную точку, и NO в противном случае.

Примеры

trian.in	trian.out
3 3 1 1 3 3 -2 -2	YES
3 4 7 -4 6 6 -20 1	NO

Комментарии

Рисунок к первому примеру из условия:



Задача G. Словарь

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

Формат входных данных

В первой строке содержится единственное целое число N — количество английских слов в словаре. Далее следует N описаний. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отведённый пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100 000.

Формат выходных данных

Выведите соответствующий данному латинско-английский словарь, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т.д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

Примеры

<code>stdin</code>	<code>stdout</code>
3 apple - malum, pomum, popula fruit - bacca, bacca, pomum punishment - malum, multa	7 bacca - fruit bacca - fruit malum - apple, punishment multa - punishment pomum - apple popula - apple popum - fruit

Комментарии

«Лексикографический порядок» означает, что слова идут по алфавиту. Иначе говоря, если у двух слов A и B несколько первых символов совпадают, то раньше идет то, у которого первая буква после общей части идет в алфавите раньше (например, слова `solution` и `solve` идут именно в таком порядке, так как первые 3 буквы в этих словах совпадают, а 4-я буква `u` в слове `solution` идет по алфавиту раньше буквы `v` слова `solve`). Если слово A является началом слова B , то раньше идет слово A (например, сначала идет слово `school`, а затем слово `schoolboy`).

Задача Н. Пешки

Имя входного файла: `pawns.in`
Имя выходного файла: `pawns.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

В первом классе Глеб увлекался шахматами. К тому моменту он знал только лишь как ходит пешка: она может бить по диагонали влево-вверх и вправо-вверх, и ходить на клетку вверх только если та клетка не занята другой фигурой. О том, что пешка может превращаться в ферзя Глеб не подозревает. Поэтому он придумал свой вариант шахмат.

Игра идёт на доске с N строками и M столбцами ($1 \leq N \leq 100$, $1 \leq M \leq 100$) по следующим правилам. В нижней строке, имеющей номер 1, стоят P белых пешек, белых фигур на доске больше нет. На остальной части доски стоят разные чёрные фигуры (их названия Глеб не знает). Ходят только белые, их цель — побить все чёрные фигуры.

Как и в настоящих шахматах, если пешка Глеба бьёт чёрную фигуру, то она становится на её место, а побитая фигура убирается с доски. Считается, что Глеб выиграл, если он сумел побить белыми пешками все чёрные фигуры, в противном случае он проиграл. Помогите ему по заданной конфигурации всех фигур определить, сможет ли он выиграть, и, в случае успеха, выведите правильную последовательность ходов белых пешек.

Формат входных данных

Сначала вводятся четыре целых числа N , M , P , K ($1 \leq N \leq 100$, $1 \leq M \leq 100$, $0 \leq P \leq M$, $1 \leq K \leq 1000$, $K \leq (M - 1)N$). Далее записано P различных чисел — номера столбцов p_j ($1 \leq p_j \leq M$), в которых стоят белые пешки. Далее идут K различных пар целых чисел — координаты (строки и столбцы) чёрных фигур r_i, c_i ($2 \leq r_i \leq N$, $1 \leq c_i \leq M$).

Формат выходных данных

Если пешки не смогут съесть все фигуры, выведите единственное слово NO.

В противном случае в первую строку выведите YES, вторая строка должна содержать суммарное число перемещений C , последующие C строк — описание ходов пешек, по одному ходу на каждую строку. Каждый ход задаётся двумя координатами r, c пешки (номерами строки и столбца), которая будет ходить, и символом m , принимающем три значения: L, R, F — побить вперед и влево, побить вперед и вправо, сделать шаг вперед соответственно. Данные о ходе следует выводить разделёнными одним пробелом, сначала координаты, потом тип хода.

Если последовательностей ходов несколько, выведите любой из них. Обратите внимание, что минимизировать количество перемещений не требуется.

Примеры

<code>pawns.in</code>	<code>pawns.out</code>
2 2 2 1 1 2 2 2	YES 1 1 1 R
3 3 2 2 1 3 3 1 3 3	NO

Задача I. Трамвай

Имя входного файла: tram.in
Имя выходного файла: tram.out
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Вася идет из школы домой вдоль проспекта, по которому ходят трамваи. Мама считает, что ему после школы полезно дышать свежим воздухом, поэтому настаивает, чтобы не менее K метров он прошел пешком. Вася при этом хочет попасть домой как можно быстрее (обязательно выполнив требование мамы).

Вдоль проспекта расположено N трамвайных остановок, которые находятся в точках a_1, a_2, \dots, a_N (все координаты задаются в метрах). Школа находится около 1-й остановки, а дом — около остановки номер N . Мальчик идет пешком со скоростью v метров в минуту. Трамвай едет со скоростью w метров в минуту (временем стоянки трамвая на остановках пренебрежем). В нулевой момент времени и далее с интервалом T минут от первой остановки в сторону Васиного дома отправляются трамваи. Вася выходит из школы также в момент времени 0. Сесть в трамвай и выйти из него можно только на остановке. При этом, если Вася приходит на остановку раньше трамвая, на который хочет сесть, то ему придется подождать, пока тот не подъедет. Вася идет пешком и едет на трамвае только в направлении от школы к дому.

Напишите программу, которая определит, когда Вася сможет оказаться дома.

Формат входных данных

Сначала вводится число N — количество остановок ($1 \leq N \leq 2000$). Далее заданы координаты остановок a_1, a_2, \dots, a_N ($0 \leq a_1 < a_2 < \dots < a_N \leq 10^9$). Далее вводится интервал движения трамваев T ($1 \leq T \leq 2000$). Затем расстояние, не меньше которого Вася должен пройти пешком K ($0 \leq K \leq 2000$). Затем заданы скорости Васи v и трамвая w ($1 \leq v \leq w \leq 10\,000$). Все вводимые числа целые. K не превышает длины пути от школы до дома.

Формат выходных данных

В первую строку выведите не менее чем с пятью знаками после десятичной точки одно число — минимальное время, когда Вася сможет оказаться дома, пройдя пешком не менее K метров. Далее нужно вывести информацию о пути Васи. Занумеруем промежутки между соседними остановками числами от 1 до $N - 1$ (то есть промежуток между первой и второй остановками имеет номер 1, между второй и третьей — 2 и так далее). Следующая строка должна содержать количество промежутков, пройденных Васей пешком. Далее выведите номера этих промежутков в возрастающем порядке.

Примеры

tram.in	tram.out
3 0 10 30 5 10 1 5	16.000000 1 1
4 0 3 8 11 1 6 1 3	7.666667 2 1 3

Задача J. Путешествия в реальности

Имя входного файла: `alternative.in`
Имя выходного файла: `alternative.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Каждый раз, когда в мире происходит значимое событие, наша реальность разветвляется на несколько — в зависимости от исхода этого события. После этого существует уже не только наша основная реальность, но и ответвившиеся от неё в моменты появления разных исходов.

Однажды один архимаг решил сделать мир лучше. Такая грандиозная задача не под силу одному архимагу, поэтому он решил найти самого себя ещё в K реальностях и выполнить эту задачу вместе. Проведённое теоретическое исследование показало, что, кроме реальности, в которой находится именно он, существует ещё $N - 1$ реальностей. Для удобства они были занумерованы числами от 1 до N , при этом его собственная реальность имеет номер 1, а посетить ему необходимо реальности с номерами $2, 3, \dots, K + 1$.

Как уже говорилось, каждая реальность когда-то ответвилась от некоторой другой, за исключением одной Начальной реальности, которая существовала всегда (её номер может оказаться каким угодно; считается, что она появилась в момент времени 0). Исследование показало, что реальность с номером i ответвилась от реальности с номером P_i в момент времени T_i . Из каждой реальности с номером i архимаг может переместиться

- в любую ответвившуюся от неё, то есть в любую j , такую что $P_j = i$;
- в P_i , если i — не Начальная реальность.

Другими словами, возможны лишь переходы вида $i \leftrightarrow P_i$. На каждый такой переход в любую сторону архимаг затрачивает $T_i - T_{P_i} > 0$ условных единиц энергии.

Требуется найти минимальное количество энергии, которое потребуется архимагу, чтобы, начав в реальности с номером 1, посетить все реальности с номерами от 2 до $K + 1$ (в любом порядке) и затем вновь вернуться в 1. Любую реальность при этом разрешается посещать сколько угодно раз.

Формат входных данных

Сначала вводятся два целых числа N и K ($0 \leq K < N \leq 100\,000$): количество доступных реальностей и количество реальностей, которые необходимо посетить. Далее идёт N пар целых чисел, i -я пара — это P_i и T_i ($1 \leq P_i \leq N$, $0 \leq T_i \leq 10^6$; для Начальной реальности $P_i = T_i = 0$).

Гарантируется, что ответвившаяся реальность появилась строго позже породившей ($T_i > T_{P_i}$), и что маг может при желании добраться до любой из N реальностей.

Формат выходных данных

Выведите единственное число E — минимальную возможную энергию, которая потребуется архимагу для путешествия.

Примеры

<code>alternative.in</code>	<code>alternative.out</code>
5 2	30
4 2	
4 6	
1 9	
0 0	
1 7	