

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 ($1 \leq N \leq 100$) 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 ($1 \leq N \leq 100$, $0 \leq A \leq 100$) — количество задач и исходное умение. Далее идут N пар целых чисел a_i , b_i ($1 \leq a_i \leq 100$, $1 \leq b_i \leq 100$) — 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 может 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 майку и штаны 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 ($1 \leq N \leq 100\,000$) маек и M ($1 \leq M \leq 100\,000$) штанов, про каждый элемент известен его цвет (целое число от 1 до 10 000 000). Помогите Глебу выбрать одну майку и одни штаны так, чтобы разница в их цвете была как можно меньше.

Формат входных данных

Сначала вводится информация о майках: в первой строке целое число N ($1 \leq N \leq 100\,000$) и во второй N целых чисел от 1 до 10 000 000 — цвета имеющихся в наличии маек. **Гарантируется**, что номера цветов идут в возрастающем порядке (в частности, цвета никаких двух маек не совпадают).

Далее в том же формате идёт описание штанов: их количество M ($1 \leq M \leq 100\,000$) и в следующей строке M целых чисел от 1 до 10 000 000 в возрастающем порядке — цвета штанов.

Формат выходных данных

Выведите пару неотрицательных чисел — цвет майки и цвет штанов, которые следует выбрать Глебу. Если вариантов выбора несколько, выведите любой из них.

Примеры

style.in	style.out
2 3 4 3 1 2 3	3 3
2 4 5 3 1 2 3	4 3

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 занять как можно более высокое место, то есть как можно меньше 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 работа 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 как можно более высокое место, а места участников в каждом из туров не изменились. При этом никакие два участника не должны получить в одном туре одинаковые баллы.

Ваша задача — проделать за жюри такую работу.

Считается, что участник по сумме двух туров занял место A , если ровно $A - 1$ участников набрали по сумме двух туров строго больше баллов.

Формат входных данных

Сначала вводится целое число N ($1 \leq N \leq 200$) — количество участников олимпиады. 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

Сначала 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, затем аналогично N целых чисел, соответствующих расстановке баллов во втором туре. Числа разделяйте пробелами или переводами строки. Никакие два участника не должны получить одинаковые баллы в одном и том же туре. Если существует несколько способов расставить баллы требуемым образом, выведите любой из них.

Примеры

<code>juggling.in</code>	<code>juggling.out</code>
3	400 399 1
2 1 3	400 399 1
3 1 2	
3	3 2 1
2 3 1	4 3 1
3 1 2	

Комментарии

Во втором примере первый участник займет второе место, разделив баллы со вторым и уступив третьему. Можно было бы сделать и так, чтобы второй участник выступил хуже первого, но место первого от этого не изменится.

Задача F. Прямоугольники

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

Петя нарисовал на клетчатом листке бумаги красивый рисунок прямоугольной формы. Его младшему брату Васе тоже захотелось порисовать, поэтому он вырезал из того же листка бумаги другой прямоугольник. При этом он не делал лишних разрезов, то есть в результате в листке осталась прямоугольная дырка. Кроме того, линии разреза не проходили (даже частично) по границам рисунка Пети. Более того, по границам рисунка не проходили даже продолжения линий разреза.

Ваша задача – по данным о расположении рисунка и прямоугольной дырки определить, испортил ли Вася рисунок старшего брата, другими словами, есть ли на вырезанном Васей прямоугольнике хотя бы маленький фрагмент рисунка Пети.

Формат входных данных

Вам даны 8 целых чисел - $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$, где (x_1, y_1) - координаты левого нижнего угла рисунка Пети, (x_2, y_2) - координаты правого верхнего угла рисунка. Аналогично, (x_3, y_3) - координаты левого нижнего угла вырезанного Васей прямоугольника, (x_4, y_4) - координаты правого верхнего угла вырезанного прямоугольника. Гарантируется, что данные прямоугольники невырождены ($x_1 < x_2, y_1 < y_2$ и аналогичные неравенства для второго набора координат). Листок был не очень большим, поэтому каждое число по модулю не превосходит 10^4 .

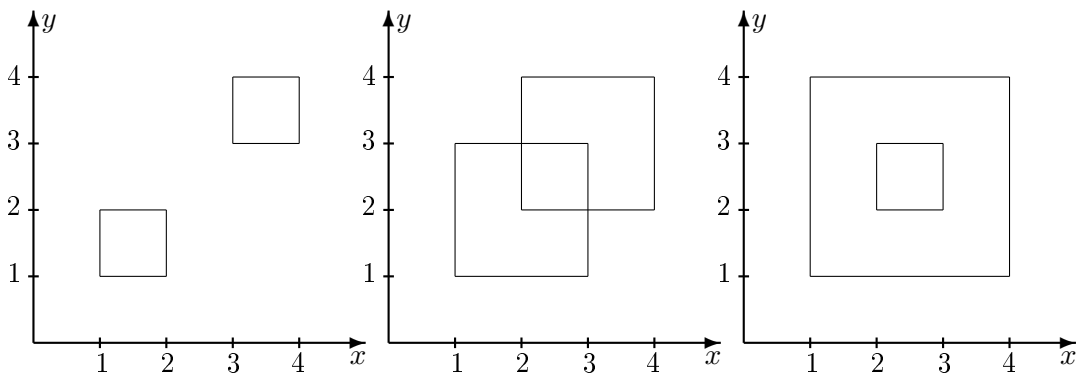
Формат выходных данных

Выведите YES, если Вася испортил рисунок, и NO в противном случае.

Примеры

<code>rects.in</code>	<code>rects.out</code>
1 1 2 2 3 3 4 4	NO
1 1 3 3 2 2 4 4	YES
1 1 4 4 2 2 3 3	YES

Комментарии



Задача G. Словарь

Имя входного файла: `voc.in`
Имя выходного файла: `voc.out`
Ограничение по времени: 1 second
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 ($1 \leq N \leq 100$) — количество английских слов в словаре. Далее следует N описаний. В первой строке каждого описания содержится английское слово. В следующей строке записано единственное число $K \geq 1$ — количество переводов. В следующих K строках приведены переводы текущего английского слова на латинский, по одному в каждой строке.

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

Формат выходных данных

Выведите соответствующий данному латинско-английский словарь в следующем формате. В первую строку запишите единственное целое число N — количество латинских слов в словаре. Далее выведите N описаний, каждое описание в отдельной строке: сначала латинское слово, затем отделённый пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого латинского слова на английский.

При этом порядок английских слов внутри перевода одного латинского может быть каким угодно. Кроме того, порядок следования латинских слов для перевода в словаре также не важен.

Примеры

<code>voc.in</code>	<code>voc.out</code>
3	7
apple	baca - fruit
3	bacca - fruit
malum	malum - apple, punishment
popum	multa - punishment
popula	popum - apple
fruit	popula - apple
3	popum - fruit
baca	
bacca	
popum	
punishment	
2	
malum	
multa	

Задача Н. Пешки

Имя входного файла: `pawns.in`
Имя выходного файла: `pawns.out`
Ограничение по времени: 1 second
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-наверх и вправо-наверх, и ходить на клетку вверх только если та клетка не занята другой фигурой. 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, стоит 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, цель — достичь хотя бы одной пешкой самой верхней строки, имеющей номер N (Глеб слышал, что в этой ситуации из пешки можно сделать ферзя, а с такой силой он безусловно сможет побить все остальные 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 достичь хотя бы одной пешкой самой верхней строки, в противном случае он проиграл. Помогите ему по заданной конфигурации всех фигур определить, сможет ли он выиграть.

Формат входных данных

Сначала вводятся четыре целых числа N , M , P , K ($1 \leq N \leq 100$, $1 \leq M \leq 100$, $0 \leq P \leq M$, $1 \leq K \leq (N - 1)M$). 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 — номера строк и столбцов чёрных фигур r_i , c_i ($2 \leq r_i \leq N$, $1 \leq c_i \leq M$).

Формат выходных данных

Если хотя бы одна пешка сможет достичь последнего ряда, выведите YES, в противном случае выведите NO.

Примеры

<code>pawns.in</code>	<code>pawns.out</code>
3 3 2 3 1 3 2 2 3 1 3 3	YES
4 4 2 4 1 4 3 1 3 2 4 2 4 4	NO