

Задача 1. Бассейны и трубы

Входной файл	pools.in
Выходной файл	pools.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_1.pas

В новом аквапарке есть N бассейнов, заполненных водой. Некоторые пары бассейнов соединены трубами, всего труб M . Система труб построена так, чтобы из любого бассейна в любой другой можно было при необходимости перегнать воду, возможно, воспользовавшись другими бассейнами как промежуточными. На каждой трубе стоит программируемый насос, который способен перекачивать в любую сторону произвольный заданный поток воды. Кроме того, в некоторые бассейны дополнительно втекает вода с известной скоростью, а из некоторых вода вытекает — тоже с известной скоростью.

Ваша задача — написать программу, которая отрегулирует насосы так, чтобы уровень воды в бассейнах поддерживался постоянным.

Формат входных данных

В первой строке указано количество бассейнов N ($1 \leq N \leq 100$). Далее следуют N строк, содержащих информацию о бассейнах. Для каждого бассейна, начиная с 1-го и кончая N -м, указана разница между дополнительным втекающим и вытекающим потоком — целое число, не превышающее по модулю 30 000 (положительное число соответствует втеканию, отрицательное — вытеканию).

На следующей строке входного файла находится число труб M ($0 \leq M \leq 4950$). В последующих M строках находится описание труб: для каждой трубы указаны два числа — номера бассейнов, которые труба соединяет. Между каждой парой бассейнов есть не более одной трубы, трубы не могут соединять бассейн с самим собой.

Формат выходных данных

Если решение существует, выведите в выходной файл M строк, по числу на строку. Для каждой трубы в том же порядке, как во входном файле, должно быть указано целое число — поток воды через неё. Положительным числом обозначается поток от бассейна, указанного первым во входном файле, ко второму; отрицательным — обратный поток. Потоки в выходном файле не должны превышать 2 000 000 000.

Если возможно несколько решений, можно выводить любое из них.

В случае, когда решения не существует, выходной файл должен содержать одну строку 'Impossible'.

Пример

Входной файл	Выходной файл
4 -100 -100 100 200 4 1 2 4 3 1 3 4 1	Impossible
4 -100 -100 100 100 4 1 2 4 3 1 3 4 1	100 0 -100 100

Задача 2. Батон

Входной файл	baton.in
Выходной файл	baton.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_2.pas

Несколько школьников решили отпраздновать Новый Год вместе. Для празднования они закупили гору продуктов, и в том числе батон колбасы. Количество еды было настолько велико, что об этом батоне вспомнили только через два с лишним часа после боя курантов. И, конечно, в честь столь знаменательного момента времени школьники решили как можно быстрее съесть этот батон. Но на этом пути их поджидала одна трудность: батон надо разрезать на N (по числу школьников) равных частей, причём как можно быстрее, чтобы не упустить момент. Для этой цели нашёлся очень длинный и острый нож, который смог бы при необходимости разрезать даже N батонов колбасы за один раз. Кроме того, нашлась линейка, с помощью которой можно отмерять любую наперёд заданную часть батона, выбирая место будущего разреза с достаточной точностью. Батон колбасы оказался настолько тонким и длинным, что резать его имело смысл только перпендикулярно его оси.

Конечно, школьники быстро сообразили, что для ускорения процесса можно резать несколько кусков батона одновременно. Батон колбасы оказался прямым и негнущимся, поэтому сэкономить число действий за счёт складывания батона перед разрезанием нельзя. Им осталось определить минимальное количество движений ножом, с помощью которых можно осуществить их намерения. Помогите им в этом.

В процессе разрезания школьник может приложить некоторые имеющиеся у него куски батона колбасы друг к другу как он хочет (напомним, что батон очень длинный и тонкий, поэтому имеет смысл прикладывать куски только параллельно друг другу), и одним движением ножа разрезать все куски. Изначально у школьника есть один кусок — сам батон; в конце у него должно быть N одинаковых по длине кусков (каждый длиной в $1/N$ начального батона).

Например, разрезать батон на 6 частей ($N = 6$) школьник может всего за 3 действия ножом:

1. Отмерить $1/3$ длины всего батона и в этом месте разрезать батон.
2. Большой из получившихся кусков разрезать пополам (в итоге получаются три куска равной длины).
3. Приложив их вплотную друг к другу, выровнять концы и разрезать все три куска посередине, в результате чего получится шесть одинаковых кусков.

Чтобы разрезать батон на 5 равных частей ($N = 5$), также необходимо 3 действия ножом, например:

1. Отмеряем $2/5$ длины батона и разрезаем батон в этом месте.
2. Получившиеся куски складываем вместе так, чтобы середина меньшего куска находилась против трети большего, и разрезаем их так, чтобы меньший разрезать пополам, а больший — в отношении $1 : 2$. Получаем три куска длиной в $1/5$ всего батона и один — длиной $2/5$.
3. Большой из имеющихся кусков разрезаем пополам и получаем пять одинаковых кусков.

Формат входных данных

Во входном файле находится одно число N ($1 \leq N \leq 2\,000\,000\,000$).

Формат выходных данных

Выведите в выходной файл одно число — минимальное количество движений ножом.

Пример

Входной файл	Выходной файл
6	3
5	3

<i>Входной файл</i>	obfuscate.in
<i>Выходной файл</i>	obfuscate.out
<i>Ограничение по времени</i>	1 сек
<i>Максимальный балл за задачу</i>	100
<i>Пример названия программы</i>	p99_3.pas

Напишите обфускатор для этого языка. А именно, ваша программа должна сделать следующее:

Синтаксис языка NN# следующий:

Формат входных данных

Во входном файле находится синтаксически корректная программа на языке NN#. Размер программы не превышает 32 Кб. Число различных переменных в программе не превышает 1000, длина имени каждой не больше 32 символов. Все символы во входном файле имеют ASCII-коды от 32 до 127 включительно (кроме переводов строк, которые, как уже было сказано, задаются последовательностью CR+LF).

Формат выходных данных

Выведите в выходной файл программу, полученную из исходной в результате необходимых преобразований. Допускается изменение регистра символов, если оно не влияет на работу программы.

Пример

Входной файл	Выходной файл
<pre>ProGram_{ rEad/**/X; print_if+20*("This/**/_is_some\"text\""+ _+(-10)); print x; if_(a<b){ _ _ _if_(B<a)_ {/**/test _ _ _ _ _print_ "Wow!_That's_a_Wonder"; _ _ _} }_/*_this_is a_comment*/ a=b+c+d+a("text"/10,10+10) _ _ _ _ _-20*"-_" "; print_(a+x); }</pre>	<pre>proGraM{rEad_a;print_B+20*("This/**/_is_ some\"text\""+(-10));print_a;if(c<d){ if(d<c){print"Wow!_That's_a_Wonder";}} c=d+e+f+a("text"/10,10+10)-20*"-_" ";print(c+a);}</pre>

Примечание: Символ '_' обозначает пробел.

Примечание: Для удобства чтения пример выходного файла разбит на строки. На самом деле выходной файл в примере должен состоять из *двух* строк с единственным разрывом строки — там, где в примере находится символ ¶ (сам этот символ, конечно, не должен присутствовать в выходном файле).

Задача 4. А ты купи коня!

Входной файл	horses.in
Выходной файл	horses.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_4.pas

Вашего друга, который до недавних пор работал кондуктором в маршрутке, повысили, и с этих самых недавних пор он является главным экономистом в фирме, владеющей этими маршрутками. К сожалению, он очень быстро обнаружил, что его знаний недостаточно для такой работы, и стал срочно навёрстывать упущенное. К счастью, это оказалось не столь сложно, благо изучать ему требовалось не всю экономику, а только ту её часть, которая имеет отношение к рынку маршруток в крупных городах.

Среди множества моделей, предложенных экономистами для описания различных видов рынков, вашему другу больше всего понравилась знаменитая «модель конного рынка» Е. фон Бём-Баверка. Эта модель состоит в следующем: в базарный день на рыночной площади собрались N продавцов и M покупателей. У каждого продавца есть одна единица товара (принято говорить о конях); каждый покупатель хочет купить одного коня. С точки зрения покупателей, все продавцы и все кони абсолютно одинаковы, т. е. покупателям всё равно, у кого покупать (значение имеет только цена). Аналогично, с точки зрения продавцов все покупатели одинаковы: продавцам всё равно, кому продавать (опять-таки, значение имеет только цена). Но субъективные оценки товара различаются:

1. Каждый продавец заранее определил для себя цену, дешевле которой он продавать ни в коем случае не будет: i -й продавец готов продавать коня по цене s_i и выше.
2. Аналогично, каждый покупатель заранее определил для себя цену, дороже которой он покупать ни в коем случае не будет: i -й покупатель готов покупать коня по цене b_i и ниже.

Вдобавок к этим условиям, Е. фон Бём-Баверк справедливо полагает, что каждый продавец будет по возможности запрашивать более высокую цену, в то время как каждый покупатель будет стараться купить как можно дешевле. Далее у Бём-Баверка следуют некоторые рассуждения, которые определяют, какие же конкретно сделки произойдут. По мнению вашего друга, эти рассуждения эквивалентны следующим двум условиям:

3. Все сделки, которые будут совершены, будут совершены по одной цене.
4. Количество продавцов, согласных продавать по этой цене, должно быть равно количеству покупателей и определяет количество сделок.

Тем не менее, эти правила всё ещё не определяют однозначно цены, по которым будут совершены сделки. Ваш друг попросил вас написать программу, которая определит нижнюю и верхнюю границу диапазона цен, в котором могут быть совершены сделки.

Примечание: Вы не ошиблись: вы действительно находитесь на олимпиаде по информатике, а не по экономике :). Поэтому жюри никоим образом не интересуется, не собирается обсуждать и не гарантирует корректность, правильность, реальность и т. д. предложенной модели. Это — задача по информатике, а не по экономике.

Формат входных данных

В первой строке входного файла находятся два целых числа N и M ($1 \leq N, M \leq 1000$) — количество продавцов и покупателей соответственно. На второй строке находятся N целых чисел s_1, s_2, \dots, s_N , а на третьей строке — M целых чисел b_1, b_2, \dots, b_M ; эти числа положительны и не превосходят 2 000 000 000.

Формат выходных данных

Если в соответствии с пп. 1—4 может произойти хотя бы одна сделка, выведите в выходной файл два числа: сначала нижнюю границу диапазона, потом — верхнюю. Если не произойдёт ни одной сделки, выведите в выходной файл одну строку 'No solution' (без кавычек).

Пример

Входной файл	Выходной файл
4 3 150 110 200 1000 190 100 140	140 150
1 1 100 90	No solution
2 1 150 150 100	No solution

Задача 5. Строительство в городе

Входной файл	build.in
Выходной файл	build.out
Ограничение по времени	4 сек
Максимальный балл за задачу	100
Пример названия программы	p99_5.pas

Новейшие веяния достигли и Солнечного города. Один местный бизнесмен решил построить в городе современный торгово-развлекательный комплекс. Ведущие архитекторы города уже разработали проект, теперь осталось только определить, где он будет размещён. Конечно, предприниматель хотел бы, чтобы комплекс был построен как можно ближе к центру города, но городская администрация оказалась категорически против сноса любых зданий в городе. Поэтому теперь перед предпринимателем встала задача: найти находящийся ближе всего к центру города свободный участок достаточного для строительства размера. Напишите программу, которая решит эту задачу.

Солнечный город, как известно, застроен круглыми домами. Естественно, дома не пересекаются, но некоторые могут касаться. Торгово-развлекательный комплекс тоже будет круглым и тоже не должен пересекаться с уже существующими зданиями (касания допустимы). Расстояние от здания до центра города понимается как расстояние от центра этого здания до центра города.

Формат входных данных

На первой строке входного файла находятся два числа: целое N ($1 \leq N \leq 800$) и вещественное R ($0 < R \leq 10^6$) — количество уже существующих зданий и радиус торгово-развлекательного комплекса.

Далее следуют N строк, на i -й из которых находятся три числа x_i , y_i и r_i — координаты центра и радиус i -го здания. Координаты не превосходят 10^6 по модулю, радиусы положительны и не превосходят 10^6 .

Система координат введена таким образом, что центр города имеет координаты $(0, 0)$.

Формат выходных данных

В выходной файл выведите два числа: координаты центра торгово-развлекательного комплекса. Выданная точка должна удовлетворять следующим требованиям:

- 1) для каждого i расстояние от выданной точки до центра i -го здания должно быть больше, чем $r_i + R - 10^{-3}$;
- 2) расстояние от выданной точки до начала координат должно отличаться от оптимального не более чем на 10^{-3} .

Пример

Входной файл	Выходной файл
2 1.0 -1 0 1.0 1 0 1	0 1.7320508075688772

Задача 6. Кофе

Входной файл	coffee.in
Выходной файл	coffee.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_6.pas

В Научно-образовательном центре ИПФ РАН недавно заново покрасили стены. Всё было бы ничего, если бы на этаж ниже не стоял кофейный аппарат: особую опасность для недавно покрашенных стен представляют школьники, пьющие кофе, поскольку иногда они разливают кофе и он забрызгивает стены, находящиеся от школьника на расстоянии, меньшем некоторого заданного R (если расстояние до стены равно R , она не пачкается). Для борьбы с этой напастью в одной из секретных лабораторий ИПФ РАН разрабатывается система, которая в момент разливания школьником кофе регистрирует его координаты (x, y) и радиус разлёта кофе R . Предполагается, что она будет устанавливаться во всех свежепокрашенных комнатах, а данные наблюдений будут записываться в единый журнал. Каждая комната считается прямоугольником ненулевой площади со сторонами, параллельными осям координат.

Вашей задачей в данном проекте является разработка кода, который будет определять, какие случаи разливания кофе, зафиксированные в журнале, завершились запачкиванием стен.

Формат входных данных

Во входном файле приведено содержание журнала разливания кофе. В первой строке указано число записей в журнале N ($1 \leq N \leq 20$). Во второй и последующих строках приведены сами записи, по одной записи на строку. Каждая запись представляет собой 7 чисел $x_1, y_1, x_2, y_2, R, x, y$, разделённых пробелами. Здесь $(x_1, y_1), (x_2, y_2)$ — координаты двух противоположных углов комнаты, в которой зафиксировано данное разливание кофе, R — радиус разлёта кофе, (x, y) — координаты точки внутри комнаты, в которой кофе был разлит. Все координаты во входном файле целые и не превосходят по модулю 10 000, радиус R тоже целый, $0 \leq R \leq 10\,000$.

Формат выходных данных

В выходом файле должно быть N строк. На i -й строке должно быть выведено 'yes', если при i -м разливании кофе испачкало стену, иначе должно быть выведено 'no'.

Пример

Входной файл	Выходной файл
3	no
-2 1 10 9 1 0 2	no
3 5 16 -3 2 10 0	yes
3 -6 5 20 2 4 13	