

Задача А. Комнаты и переходы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В подземелье есть $(n + 1)$ комнат, последовательно соединенных n переходами. Комнаты пронумерованы от 0 до n , а переходы — от 1 до n , и i -й переход соединяет комнаты $(i - 1)$ и i .

Каждый переход оснащен охранным устройством одного из двух типов. Охранное устройство первого типа при переходе проверяет наличие пропуска определенного цвета, и запрещает проход, если пропуск такого цвета недействителен. Охранное устройство второго типа никогда не запрещает проход, но при проходе пропуск определенного цвета аннулируется и становится недействительным.

В начале вы обладаете всеми возможными пропусками и находитесь в комнате s . Для каждого возможного s от 0 до $(n - 1)$ выведите, как много комнат можно пройти в направлении комнаты n .

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 500000$) — количество переходов.

Во второй строке дано n целых чисел a_i ($1 \leq |a_i| \leq n$). Число $|a_i|$ — цвет пропуска, с которым работает i -й переход. Если $a_i > 0$, то i -й переход — первого типа, а если $a_i < 0$ — второго типа.

Формат выходных данных

Выведите n целых чисел через пробел: ответ на задачу для каждого возможного s от 0 до $(n - 1)$.

Примеры

стандартный ввод	стандартный вывод
6 1 -1 -1 1 -1 1	3 2 1 2 1 1
7 2 -1 -2 -3 1 3 2	4 3 3 2 3 2 1

Задача В. Переставить столбцы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Есть клетчатое поле размера $2 \times n$, некоторые клетки которого помечены. Требуется переставить его столбцы так, чтобы все помеченные клетки образовывали бы связную область.

Помеченные клетки образуют связную область, если из любой из них можно добраться до любой другой, совершая перемещения вверх, вниз, влево или вправо (не обязательно одинаковые), и все клетки в пути тоже являются помеченными.

Формат входных данных

Входные данные содержат описание поля: две строки одинаковой длины n ($1 \leq n \leq 1000$), состоящие из символов «.» и «#», обозначающих соответственно непомеченные и помеченные клетки. Гарантируется, что хотя бы одна клетка помечена.

Формат выходных данных

В первой строке выведите «YES» или «NO», в зависимости от того, можно ли переставить столбцы требуемым образом.

В случае положительного ответа далее выведите две строки, состоящие из символов «.» и «#» — получившееся после перестановки столбцов поле.

Если существует несколько возможных ответов, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
#. .# .# .#	YES ##. . .##.
. .## ##. .	NO

Задача С. Прыжки по окружности

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На окружности расположено p полей, пронумерованных от 0 до $(p - 1)$ так, что из поля 0 можно перейти в поле 1, из поля 1 в поле 2, и т. д., а из поля $(p - 1)$ можно перейти в поле 0.

Фишка изначально расположена на поле 0. Она начинает бегать по окружности, делая сперва ход на 1 поле вперед, потом на 2, потом на 3, и т. д.

Сколько уникальных полей (включая начальное) посетит фишка за n ходов?

Формат входных данных

Входные данные содержат два целых числа p и n ($1 \leq p \leq 10^7, 0 \leq n \leq 10^{18}$) — количество полей на окружности и количество ходов.

Формат выходных данных

Выведите целое число — количество уникальных полей, которое посетит фишка.

Примеры

стандартный ввод	стандартный вывод
3 10	2
5 3	3
8 1000000000000000000	8

Задача D. Разделение страны

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В одной стране есть n городов, между которыми проведено $(n - 1)$ дорог, причем из любого города можно добраться до любого другого. В преддверии выборов политологи спрогнозировали q вариантов развития событий. В каждом из этих вариантов предполагается, что жители некоторых городов будут поддерживать одного кандидата (назовем такие города красными), жители некоторых других городов будут поддерживать другого кандидата (назовем такие города синими), а жители остальных городов равнодушны к политике, и им все равно.

Для каждого из вариантов вам необходимо определить, можно ли будет перекрыть некоторые дороги так, чтобы из любого красного города можно было добраться до любого красного, из любого синего города можно было добраться до любого синего, но из любого красного города нельзя было добраться до любого синего.

Формат входных данных

В первой строке дано целое число n ($2 \leq n \leq 200000$) — количество городов.

В следующих $(n - 1)$ строках даны два целых числа u_i, v_i ($1 \leq u_i, v_i \leq n$) — номера городов, соединенных дорогами.

В следующей строке дано целое число q ($1 \leq q \leq 200000$) — количество прогнозов.

В каждой из следующих q строк дано описание очередного прогноза. Оно начинается с двух целых чисел r_j и b_j ($1 \leq r_j, b_j < n, r_j + b_j \leq n$) — количеств красных и синих городов соответственно. Затем даны r_j целых чисел — номера красных городов. Затем даны b_j целых чисел — номера синих городов. Все номера городов в пределах одного прогноза различны.

Сумма всех r_j и b_j во всех прогнозах не превышает 200000.

Формат выходных данных

Для каждого прогноза выведите ответ на него — «YES» или «NO» — на отдельной строке.

Пример

стандартный ввод	стандартный вывод
7	YES
1 2	NO
1 3	NO
2 4	YES
2 5	YES
3 6	YES
3 7	
6	
2 2 4 5 6 7	
2 2 4 6 5 7	
2 1 4 5 2	
2 1 4 5 1	
1 1 1 2	
6 1 1 2 3 4 5 6 7	

Задача E. Third-Party Software - 2

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Павел разрабатывает еще одну игру. Для этого ему снова очень нужны функции, доступные в сторонней библиотеке, слишком известной, чтобы ее называть. Всего есть m функций, пронумерованных от 1 до m , и известно, что i -я версия библиотеки содержит функции от a_i до b_i включительно.

Библиотека не бесплатная, а Павлу нужны все функции. Какое минимальное количество версий надо приобрести, чтобы можно было пользоваться всеми функциями?

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n \leq 200000, 1 \leq m \leq 10^9$) — количество версий библиотеки и количество функций.

В каждой из следующих n строк дано два числа a_i и b_i ($1 \leq a_i \leq b_i \leq m$) — промежуток номеров функций, доступных в версии i .

Формат выходных данных

В первой строке выведите «YES» или «NO», в зависимости от того, удастся ли приобрести версии библиотеки так, чтобы можно было пользоваться всеми функциями, или нет.

В случае положительного ответа выведите еще две строки. Во второй строке выведите целое число k — минимальное количество версий библиотеки, которые нужно приобрести. В третьей строке выведите k различных целых чисел — номера версий, которые нужно приобрести.

Если существует несколько возможных ответов, разрешается вывести любой.

Примеры

стандартный ввод	стандартный вывод
4 8 1 2 3 4 5 6 7 8	YES 4 1 2 3 4
4 8 1 5 2 7 3 4 6 8	YES 2 1 4
3 8 1 3 4 5 6 7	NO

Задача F. Friendly Fire

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Никита — поклонник продукции Volvo Software. Не так давно эта компания выпустила игру, слишком известную, чтобы ее называть.

У противника Никиты есть n существ, и i -е из этих существ обладает атакой a_i и имеет h_i хит-поинтов. У Никиты есть заклинание Friendly Fire. Это заклинание применяется на двух вражеских существ, и эти существа одновременно атакуют друг друга один раз. Если значение атаки атакующего существа больше или равно количеству хит-поинтов у цели, цель умирает.

Помогите Никите выбрать такие две цели для Friendly Fire, чтобы суммарная атака вражеских существ стала как можно меньше.

Формат входных данных

В первой строке дано целое число n ($2 \leq n \leq 300000$) — количество существ у противника.

В каждой из следующих n строк даны два целых числа a_i и h_i ($1 \leq a_i, h_i \leq 10^9$) — атака и количество хит-поинтов i -го существа.

Формат выходных данных

В первой строке выведите целое число — максимально возможное уменьшение суммарной атаки вражеских существ.

Во второй строке выведите два различных целых числа от 1 до n — номера существ, на которых следует применить заклинание.

Если возможных ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
3 4 3 3 1 5 4	9 3 1
2 1 2 4 8	1 2 1
2 1 2 1 2	0 1 2

Задача G. Акинатор

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Однажды хакер Дмитрий открыл для себя интересную игру — Акинатор. Правила игры очень просты: игрок загадывает какого-нибудь известного человека, а Акинатор пытается угадать этого человека за ограниченное количество вопросов. Каждый ход Акинатор спрашивает игрока: «Этот человек — i_1 или i_2 или ... или i_m ?» (здесь m — это количество человек, про которых был задан вопрос, оно может меняться от вопроса к вопросу). Игрок может ответить «Да» или «Нет», а Акинатор будет использовать полученную информацию. Как только Акинатор гарантированно угадал задуманного человека, он говорит ответ и выигрывает. Если же Акинатор не может уложиться в заданное число вопросов, он проигрывает.

Акинатор должен угадать задуманного человека за k вопросов. Заранее известно, что Дмитрий задумал одного из n человек: $1, 2, \dots, n$ с вероятностями $p_1, p_2, \dots, p_n$. Определите, сможет ли Акинатор гарантированно угадать человека за k вопросов, и если да, какое минимальное среднее число вопросов ему для этого понадобится.

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq k \leq n \leq 100$) — количество человек, которые могли быть загаданы и количество вопросов.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^{12}$). Вероятности p_i прямо пропорциональны a_i , т. е. могут быть вычислены как $p_i = \frac{a_i}{a_1 + a_2 + \dots + a_n}$.

Формат выходных данных

Если Акинатор не сможет гарантированно угадать задуманного человека, выведите «No solution».

Иначе выведите несократимую рациональную дробь — минимально возможное среднее число вопросов, которое понадобится Акинатору, чтобы победить.

Примеры

стандартный ввод	стандартный вывод
4 1 8 1 9 2	No solution
4 2 1 2 3 4	2/1
4 3 1 2 3 4	19/10

Замечание

В третьем примере Акинатор может задать 3 вопроса. Оптимальным будет задать первый вопрос про четвертого человека. Если будет получен ответ «Нет», то следует спросить про третьего человека. Если же и в этом случае будет получен ответ «Нет», то последним вопросом нужно сделать выбор между первым и вторым человеком.

Среднее количество вопросов в этом случае равно $0.4 \cdot 1 + 0.3 \cdot 2 + 0.2 \cdot 3 + 0.1 \cdot 3 = 1.9$.

Задача Н. Недостающее число

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Томаса есть массив целых чисел $a_1, \dots, a_n$, все элементы которого различны и находятся в промежутке от 0 до n . Как можно заметить, одного из чисел от 0 до n нет в этом массиве.

Вам нужно найти это число. Вы можете не более $2 \cdot n + 19$ раз спросить, чему равен b -й бит числа a_i .

Протокол взаимодействия

Это интерактивная задача. Ваша программа должна общаться с программой жюри, используя для этого стандартные потоки ввода и вывода.

В самом начале вашей программе сообщается целое число n ($1 \leq n \leq 1000$) — размер массива.

После этого вы можете сделать не более $2 \cdot n + 19$ запросов. Чтобы сделать запрос, выведите символ «?», а затем через пробел числа i и b ($1 \leq i \leq n, 0 \leq b \leq \log_2 n$) — индекс i в массиве и номер бита, который вы хотите узнать. Биты нумеруются с нуля, начиная с младшего.

В ответ на такой запрос сообщается число 0 или 1 — значение b -го бита числа a_i .

Как только вы определите недостающее в массиве число, выведите символ «!», а затем через пробел это число. После этого ваша программа должна завершиться.

Пример

стандартный ввод	стандартный вывод
3	
	? 1 1
0	
	? 2 0
1	
	? 3 1
0	
	! 2

Замечание

Обратите внимание, что каждое выведенное вами сообщение должно завершаться переводом строки. Также после вывода каждого сообщения ваша программа должна очищать потоковый буфер, чтобы выведенная вами информация дошла до программы жюри: например, это делают вызовы «`fflush(stdout)`» или «`cout.flush()`» в C++, «`System.out.flush()`» в Java, «`Console.Out.Flush()`» в C#, «`flush(output)`» в Pascal, «`sys.stdout.flush()`» в Python.

Пустые строки в примере приведены лишь для удобства, чтобы было лучше понятно, в каком порядке выводятся сообщения. При решении задачи вам не нужно выводить пустые строки, и программа жюри тоже не будет выводить пустые строки.

Задача I. Покраска квадрата

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Есть квадрат размера $a \times a$. В его левом верхнем углу находится квадратная кисть размера $b \times b$, которой нужно будет покрасить квадрат (можно считать, что верхний левый угол размера $b \times b$ уже покрашен). Можно передвигать кисть только параллельно сторонам квадрата. Какое минимальное расстояние должен пройти центр кисти, чтобы весь квадрат полностью был покрашен?

Формат входных данных

Входные данные содержат два целых числа a и b ($1 \leq b \leq a \leq 10^6$) — стороны квадрата и кисти соответственно.

Формат выходных данных

Выведите единственное целое число — минимальное расстояние, которое должен пройти центр кисти. Гарантируется, что ответ является целым числом.

Примеры

стандартный ввод	стандартный вывод
4 2	6
4 3	3
9 3	24
1000000 1	999999999999

Задача J. Мощь темной стороны - 2

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На джедайском турнире n джедаев сражаются друг с другом. У каждого из них есть три характеристики: сила, ловкость и интеллект. Для победы одного джедая над другим необходимо, чтобы значения как минимум двух его характеристик превосходили значения аналогичных характеристик оппонента. Например, джедай с характеристиками (5, 9, 10) победит джедая с характеристиками (2, 12, 4) за счет первой и третьей характеристик.

Ситхи планируют переманить одного из джедаев на темную сторону силы. Это даст ему могущественную способность, позволяющую перед каждой битвой установить значения своих характеристик произвольным образом, главное, чтобы все значения остались бы целыми неотрицательными числами, а сумма характеристик не изменилась бы.

Для каждого джедая определите, сколько других джедаев он сможет победить, если его переманят на темную сторону.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 500000$) — количество джедаев.

В каждой из следующих n строк содержатся три целых числа a_i , b_i и c_i ($0 \leq a_i, b_i, c_i \leq 10^9$) — характеристики джедаев.

Формат выходных данных

Выведите n чисел: сколько других джедаев сможет победить i -й джедай, если его переманят на темную сторону.

Пример

стандартный ввод	стандартный вывод
4 1 3 4 2 5 9 6 10 3 5 2 3	1 3 3 2

Задача К. Сортировка колоды

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Есть колода карт, лежащая перед вами на столе рубашкой вниз. В колоде бывают карты красного, зеленого и синего цвета, и вы знаете, в каком порядке они там находятся. Вы хотите отсортировать карты в колоде таким образом, чтобы карты каждого цвета составляли бы непрерывный отрезок. При этом порядок цветов значения не имеет.

К сожалению, на столе очень мало места — хватит лишь для еще двух стопок. За одно действие вы можете взять верхнюю карту из исходной колоды и положить ее на верх одной из стопок рубашкой вниз. После того, как все карты будут определены в одну из стопок, вы кладете одну из этих стопок на другую, опять же, рубашкой вниз.

Можно ли сделать так, чтобы после описанной процедуры колода карт стала бы отсортированной?

Формат входных данных

Во входных данных дана строка, состоящая из символов «R», «G» и «B» и имеющая длину от 1 до 1000 символов. Символы в этой строке соответствуют цветам карт в колоде, начиная с верхней.

Формат выходных данных

Выведите «YES» или «NO», в зависимости от того, удастся отсортировать колоду или нет.

Примеры

стандартный ввод	стандартный вывод
RGBRGB	YES
RGBRGBRGB	NO
RBBRRB	YES

Замечание

В первом примере колода имеет вид «RGBRGB» (начиная с верхней карты). Первую, четвертую и пятую карты можно положить в первую стопку (она будет иметь вид «GRR»), а вторую, третью и шестую карты — во вторую стопку (она будет иметь вид «BBG»). После чего вторую стопку можно положить поверх первой, и колода будет отсортированной: «BBGGRR».

Задача L. Вписанная окружность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны две окружности на плоскости. Гарантируется, что они имеют ровно две точки пересечения.

В результате пересечения двух кругов образуется некоторая фигура положительной площади, состоящая из точек, принадлежащих обоим кругам.

Требуется вписать в эту фигуру окружность наибольшего возможного радиуса. Выведите координаты ее центра и ее радиус.

Формат входных данных

В первой строке даны три целых числа x_1, y_1, r_1 ($-1000 \leq x_1, y_1 \leq 1000, 1 \leq r_1 \leq 1000$) — координаты центра первой окружности и ее радиус.

Во второй строке даны три целых числа x_2, y_2, r_2 ($-1000 \leq x_2, y_2 \leq 1000, 1 \leq r_2 \leq 1000$) — координаты центра второй окружности и ее радиус.

Формат выходных данных

Выведите три вещественных числа x, y, r — координаты центра и радиус получившейся окружности. Каждое из выведенных чисел должно иметь абсолютную или относительную погрешность, не превышающую 10^{-9} .

Примеры

стандартный ввод
0 0 5 6 0 5
стандартный вывод
3.0000000000000000 0.0000000000000000 2.0000000000000000

стандартный ввод
-12 34 56 78 -90 123
стандартный вывод
13.322257821855908 -0.888444110112585 12.890601098820779

Задача M. Shlakoblock is live!

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пользователь Shlakoblock устроил среди зрителей его канала голосование, в какую игру он будет играть на следующем стриме. В голосовании представлены n игр. Каждый зритель может проголосовать за любое подмножество этих игр, но за каждую игру можно проголосовать не более одного раза. В конце Shlakoblock выберет случайный голос и будет стримить игру, за которую этот голос был отдан.

Все, кроме вас, уже проголосовали — сейчас за i -ю игру проголосовали v_i раз. Вы оцениваете удовольствие от стрима по i -й игре как p_i . За какие игры вам следует проголосовать, чтобы максимизировать математическое ожидание удовольствия от стрима?

Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 500$) — количество тестов. Далее следуют описания тестов.

В первой строке каждого теста дано целое число n ($1 \leq n \leq 1000$) — количество игр.

В каждой из следующих n строк этого теста даны два целых числа p_i и v_i ($0 \leq p_i, v_i \leq 1000$) — удовольствие от стрима по i -й игре и количество голосов, которое сейчас за нее отдано.

Гарантируется, что в каждом тесте хотя бы одно v_i положительно.

Формат выходных данных

Выведите $3t$ строк. Ответ на каждый тест должен состоять из трех строк.

В первой строке выведите несократимую рациональную дробь — максимально возможное математическое ожидание удовольствия от стрима.

Во второй строке выведите целое число k ($0 \leq k \leq n$) — количество игр, за которые надо проголосовать.

В третьей строке выведите k различных целых чисел от 1 до n — номера игр, за которые надо проголосовать.

Если существует несколько возможных ответов, разрешается вывести любой из них.

Пример

стандартный ввод	стандартный вывод
2	6/1
5	2
10 5	1 4
4 7	2555/557
6 3	3
8 2	2 3 4
2 4	
4	
1 1000	
10 100	
100 10	
1000 1	