

Задача А. Задача для третьеклассника

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Юрий учится в третьем классе, и сегодня ночью, пытаясь уснуть, он придумал следующую задачу.

Даны две плоскости, заданные уравнениями $a_1x + b_1y + c_1n + d_1 = 0$ и $a_2x + b_2y + c_2n + d_2 = 0$ (плоскости расположены в декартовой системе координат с центром в точке $O(0, 0, 0)$ и осями Ox , Oy и On). Известно, что плоскости не совпадают и не параллельны друг другу.

Рассмотрим один из меньших двугранных углов, образованных данными плоскостями. Выберем случайную плоскость $ax + by + cn + d = 0$ и пересечем ее с рассмотренным двугранным углом. Очевидно, что с вероятностью 1 в пересечении получится некоторый плоский угол. Ваша задача — найти математическое ожидание величины данного угла в радианах. Плоскость выбирается случайно и равномерно.

Юрий, будучи величайшим математиком, разумеется, решил данную задачу. А сможете ли решить ее вы?

Формат входных данных

Первая строка содержит четыре целых числа a_1, b_1, c_1 и d_1 ($-1\,000 \leq a_1, b_1, c_1, d_1 \leq 1\,000$) — коэффициенты уравнения первой плоскости.

Вторая строка содержит четыре целых числа a_2, b_2, c_2 и d_2 ($-1\,000 \leq a_2, b_2, c_2, d_2 \leq 1\,000$) — коэффициенты уравнения второй плоскости.

Гарантируется, что данные плоскости не совпадают и не параллельны друг другу.

Формат выходных данных

Выведите одно вещественное число — математическое ожидание величины полученного плоского угла в радианах. Ответ будет считаться верным, если его относительная или абсолютная погрешность не превосходит 10^{-6} .

Пример

стандартный ввод	стандартный вывод
1 2 3 4 1 1 1 1	0.3875966867

Задача В. Беспредел

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Перестановкой называется упорядоченный набор без повторений чисел $1, 2, \dots, n$.

Пусть дана перестановка p_1, p_2, \dots, p_n . Неподвижной точкой перестановки p называется такое число i , что $p_i = i$.

Математическим ожиданием дискретной случайной величины X называется число $EX = \sum_n p_n x_n$, где x_n — значения случайной величины, а p_n — распределение вероятностей случайной величины.

Число a называется пределом числовой последовательности $a_1, a_2, \dots, a_n, \dots$, если для любого $\varepsilon > 0$ существует номер N , такой что для всех $n > N$ верно $|a_n - a| < \varepsilon$. Число a обозначается как $a = \lim_{n \rightarrow \infty} a_n$.

Рассмотрим числовую последовательность $e_1, e_2, e_3, \dots, e_n, \dots$, где e_n равно математическому ожиданию количества неподвижных точек в случайной перестановке, состоящей из n элементов.

От вас требуется вычислить предел последовательности e_n , то есть $\lim_{n \rightarrow \infty} e_n$.

Формат входных данных

В данной задаче ничего не нужно считывать.

Формат выходных данных

Выведите одно вещественное число — значение предела последовательности e_n . Если же предела не существует, можете попробовать доказать это программе-чекеру.

Ответ считается правильным, если он отличается от ответа жюри не более, чем на 10^{-6} .

Замечание

В качестве примера, вычислим математическое ожидание количества неподвижных точек в случайной перестановке из двух элементов, то есть элемент последовательности e_2 .

Всего существует две перестановки из двух элементов. В первой перестановке $(1, 2)$ есть две неподвижные точки, а во второй перестановке $(2, 1)$ неподвижных точек нет. Таким образом, математическое ожидание равно $\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 0 = 1$.

Задача С. Game Theory

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Грамм — профессор в области Computer Science в академии Sakuu, и в этом семестре он ведет предмет, посвященный теории игр.

Включая Грамм, в аудитории находятся n человек. Сегодня с целью увеличить интерес студентов к предмету, Грамм решила поиграть с ними в одну увлекательную игру.

Для каждого студента Грамм выберет целое число x ($1 \leq x \leq 20$). Не зная, какое именно число выбрала Грамм, каждый студент выберет некоторое целое число y ($1 \leq y \leq 20$). После этого Грамм вычислит балл независимо для каждого студента по следующим правилам:

- Грамм даст студенту x очков.
- Студент даст Грамм y очков.
- Если число x строго больше числа y , то Грамм получит от студента дополнительные 10 очков.
- Если число x строго меньше числа y , то Грамм даст студенту дополнительные 10 очков.

Теперь Грамм хочет знать математическое ожидание количества очков, которые она сможет выиграть от всех своих студентов, если она будет выбирать число x случайно и равномерно. Так как студенты весьма умны, можете полагать, что они действуют оптимально с целью максимизировать их финальные баллы.

Обратите внимание, что во время игры, если кто-то дает кому-то некоторое количество очков, то это количество очков отнимается от счета данного человека и прибавляется к счету другого человека. Более того, суммарное количество очков, которые кто-то получил, может быть отрицательным.

Формат входных данных

В единственной строке записано число n ($1 \leq n \leq 1\,000$) — количество людей в классе, включая Грамм.

Формат выходных данных

Выведите одно число — математическое ожидание количества очков у Грамм после окончания игры. Ваш ответ будет засчитан, если абсолютная или относительная погрешность не будет превышать 10^{-4} .

Пример

стандартный ввод	стандартный вывод
1	0.00000000000000

Задача D. Black and White

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Рассмотрим игру. В ряд лежат n шариков двух цветов: черные и белые. Позиции в ряду пронумерованы от 1 до n . Вам известно только общее количество шариков (n); точное их расположение и даже количество белых шариков неизвестно.

Вы можете делать запросы вида $v u$, где $1 \leq v, u \leq n$. Если на позиции v находится чёрный шарик, а на позиции u — белый, то эти шарики поменяют местами (иначе не произойдёт ничего). При этом вам не сообщают, поменяли шарики местами или нет.

После некоторого (возможно, нулевого) числа таких запросов вы должны сделать «утверждение». «Утверждение» — это тоже две позиции v, u . Вам нужно угадать две такие позиции в ряду, что там лежат шарики одного и того же цвета.

Ваша задача — выдать последовательность запросов, за которой следует ровно одно «утверждение».

В одном тесте может быть много игр. Более того, во всех тестах, кроме первого, ровно 99 игр, причём в первой игре $n = 2$, во второй игре $n = 3$ и т. д.. В последней игре $n = 100$.

Первый тест совпадает с примером из условия. На нём проверяется только формат вывода.

На всех тестах, начиная со второго, чтобы пройти тест, ваша программа должна угадать хотя бы в 80 играх из 99.

Обратите внимание, что, хоть ваша программа и не должна выигрывать все 99 игр, *мы не гарантируем, что тесты случайные*.

Формат входных данных

В первой строке ввода записано одно число m — количество игр в этом тесте. Для всех тестов, кроме первого, $m = 99$.

В следующих строках заданы начальные расстановки шаров в ряду. Во всех тестах, кроме первого, расстановки зашифрованы. Вы не должны пытаться расшифровать эти данные (и не обязаны их считывать), однако наличие во вводе этой информации позволяет нам гарантировать, что в этой задаче заранее определённые тесты, которые не подстраиваются под вашу стратегию.

Формат выходных данных

Выведите ровно m описаний игр.

Одно описание игры состоит из $k + 1$ строк, где k — количество сделанных вашей программой запросов.

В первых k строках выводятся эти запросы (в порядке применения) в формате $? v u$ ($1 \leq v, u \leq n, v \neq u$).

В последней строке описания выводится $! v u$ ($1 \leq v, u \leq n, v \neq u$) — ваше «утверждение».

Общее количество выведенных строк (суммарно по всем играм внутри одного теста) не должно превышать $5 \cdot 10^5$.

Примеры

стандартный ввод	стандартный вывод
2	? 1 2
01	? 2 1
101	? 1 2
	! 2 1
	? 1 2
	? 3 1
	! 1 2

Замечание

В примере из условия белые шарики обозначены как 0, чёрные — как 1. Рассмотрим его подробнее.

Первая игра: $01 \rightarrow 01 \rightarrow 10 \rightarrow 01$ — мы не угадали.

Вторая игра: $101 \rightarrow 011 \rightarrow 110$ — мы угадали.

Так, на этом тесте мы угадали в одной игре из двух.

Задача E. Дороги в городе E

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Как хорошо известно, в городе «E» за его полутора тысячелетнюю историю ни разу не ремонтировали дороги. И только недавно руководство города отремонтировало некоторые из них.

Известно, что всего в городе «E» есть n перекрестков и m дорог, пронумерованных целыми числами от 1 до m , перемещаться по которым можно в обе стороны. i -я дорога соединяет перекрестки с номерами a_i и b_i .

Среди всех m дорог было отремонтировано некоторое подмножество дорог, но вам не известно, какое именно. Единственная информация, которую вы смогли получить от дорожных служб города, это то, что от любого перекрестка можно доехать до любого другого, двигаясь только по отремонтированным дорогам.

Вы — молодой предприниматель, и решили организовать службу доставки свежего сырого мяса в городе «E» (в самом городе такое мясо называют «стейками», оно пользуется большой популярностью у местных жителей). Вы уже набрали штат курьеров, однако курьеры готовы перемещаться только по отремонтированным дорогам. Теперь вам предстоит выяснить, какие дороги уже отремонтированы.

Городская администрация предоставила вам город на некоторое время, поэтому вы можете делать различные действия одного из трех типов:

1. Заблокировать дорогу с номером x . В этом случае перемещение по дороге для курьеров будет запрещено. **Исходно все дороги разблокированы.**
2. Разблокировать дорогу с номером x . В этом случае курьеры смогут двигаться по дороге x , если она отремонтирована.
3. Попробовать доставить заказ на перекресток с номером y . В этом случае один из ваших курьеров начнет двигаться с неизвестного вам перекрестка s и доставит заказ на перекресток с номером y в том случае, если существует путь по разблокированным отремонтированным дорогам от перекрестка s до перекрестка y . При этом гарантируется, что перекресток s **будет выбран заранее.**

К сожалению, город предоставлен в ваше полное распоряжение ненадолго, поэтому вы можете сделать не более $100 \cdot m$ запросов.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит единственное целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и m ($2 \leq n \leq 2000$, $n - 1 \leq m \leq 2000$) — число перекрестков и дорог в городе «E».

Каждая из следующих m строк содержит описание одной дороги. i -я из этих строк содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$) — концы i -й дороги. Гарантируется, что никакая дорога не соединяет город сам с собой, при этом возможно, что между парой различных перекрестков есть несколько дорог.

Гарантируется, что сумма n и сумма m по всем наборам входных данных не превосходит 2000.

Протокол взаимодействия

После того, как вы считали описание набора входных данных, вы можете задавать запросы. Запросы могут быть трех типов:

1. «- x » ($1 \leq x \leq m$). В этом случае дорога с номером x блокируется, если она ещё не была заблокирована.
2. «+ x » ($1 \leq x \leq m$). В этом случае дорога с номером x разблокируется. Обратите внимание, что дорога x должна быть заранее заблокирована. **Исходно все дороги разблокированы.**
3. «? y » ($1 \leq y \leq n$). В этом случае программа жюри выбирает некоторый город s . В случае, если от города s до города y можно добраться по разблокированным отремонтированным дорогам, программа жюри выведет 1, иначе программа жюри выведет 0. Обратите внимание, что город s **будет выбран до получения информации о городе y** , однако при выборе города s могут учитываться ваши предыдущие запросы.

Всего вы можете задать не более $100 \cdot m$ запросов для каждого набора входных данных.

После того, как вы нашли все отремонтированные дороги, выведите «! $c_1, c_2, c_3, \dots, c_m$ », где c_i равно 1, если дорога i отремонтирована, и 0, если дорога не отремонтирована. Этот вывод **не будет считаться** в общем числе запросов. На это программа жюри выведет 1, если ваш ответ правильный, и 0, если ответ не правильный. Если вы считали 0, то ваша программа должна немедленно завершиться, чтобы получить вердикт **Неправильный ответ**. В противном случае вы можете получить любой вердикт, так как программа продолжит чтение из закрытого потока. Если вы считали 1, то перейдите к следующему набору входных данных, или завершите программу, если такого нет.

Обратите внимание, что вам не обязательно разблокировать все дороги на момент вывода ответа. Гарантируется, что все отремонтированные дороги зафиксированы изначально и не будут меняться программой жюри в зависимости от запросов.

После вывода запроса или ответа не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт **Решение «зависло»**. Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Взломы

Вы не можете делать взломы по этой задаче.

Пример

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

Замечание

В первом наборе входных данных дорога 1 отремонтирована, а дорога 2 — нет. Для первого запроса доставки в качестве s был выбран перекресток 1, поэтому путь от перекрестка 1 до 1 есть. Для второго запроса доставки в качестве s был выбран перекресток 1, так как в города заблокирована единственная отремонтированная дорога, то пути между перекрестками 1 и 2 нет. Для третьего запроса доставки в качестве s был выбран перекресток 2, путь между перекрестками 2 и 1 есть по дороге 1, которая отремонтирована и разблокирована.

Во втором наборе входных данных для запросов доставки в качестве стартовых перекрестков были выбраны перекрестки 1, 3, 1, 2, 2, 3, 1.

Задача F. Долгая игра

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Леша очень любит играть с кубиками. Иногда он строит из них стены, иногда рисует на них буквы, а затем составляет слова, но сегодня особенный день, ведь к Леше в гости пришла Полина, поэтому Леша придумал новую игру.

Перед началом игры Леша достал из шкафа N кубиков, после чего на каждом кубике написал число от 1 до N . Разумеется, ни на каких двух кубиках Леша не написал одно и то же число.

После успешной подготовки к игре Леша надел на глаза повязку, благодаря которой он больше не видит кубики и числа, написанные на них. Далее Леша и Полина по очереди делают ходы, причем первым ходит Леша, ведь кубики принадлежат ему!

Своим первым ходом Леша случайно равномерно перемешивает все лежащие перед ним кубики и выкладывает их в ряд. После этого Полина во время своего хода сообщает ему, какие кубики лежат на своем месте. Будем говорить, что кубик с написанным на нем числом A лежит на своем месте, если слева от него находятся ровно $A - 1$ кубиков. Во время всех следующих ходов Леша не будет трогать кубики, которые лежат на своем месте.

Далее Леша снова перемешивает все кубики, которые лежат не на своих местах, а Полина сообщает ему, какие из них оказались на своем месте. Игра продолжается до тех пор, пока все кубики не окажутся на своих местах.

Перед началом игры Леше стало интересно, насколько много ходов ему придется сделать. Посчитайте математическое ожидание количества ходов Леша, с учетом первого хода.

Формат входных данных

В единственной строке записано целое число N ($1 \leq N \leq 10^6$) — количество кубиков, которые есть у Леша.

Формат выходных данных

Нетрудно показать, что ответ можно представить в виде несократимой дроби $\frac{P}{Q}$.

В качестве ответа выведите $P \cdot Q^{-1} \pmod{998\,244\,353}$.

Примеры

стандартный ввод	стандартный вывод
1	1
2	2

Замечание

В первом примере у Леша есть всего один кубик, который после его первого хода с вероятностью 1 окажется на своем месте. Значит, математическое ожидание количества ходов Леша равно 1.

Во втором примере у Леша есть два кубика. С вероятностью $\frac{1}{2}$ после первого же хода они оба окажутся на своих местах и игра закончится, и с вероятностью $\frac{1}{2}$ ни один из кубиков не окажется на своем месте, и Леша окажется в том же положении, что и в начале игры. Тогда можно понять, что вероятность того, что игра закончится ровно через k ходов, равна $\frac{1}{2^k}$. Можно показать, что математическое ожидание этой случайной величины равно 2.

Напомним, что математическое ожидание случайной величины X равно

$$\sum_i x_i \cdot P(X = x_i)$$

Здесь $P(X = x_i)$ — вероятность того, что значение случайной величины равно x_i , а x_i — все возможные значения случайной величины.

Также напомним, что $Q \cdot Q^{-1} \equiv 1 \pmod{998\,244\,353}$.

Задача G. Fakes and Shidget

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Павел очень любит игру Fakes and Shidget. Суть игры заключается в следующем. Игрок встречается случайного персонажа из n имеющихся персонажей. Вероятности встретить каждого персонажа равны. Каждый из персонажей предлагает игроку ровно два квеста. Первый квест i -го персонажа требует a_i минут на его выполнение и приносит игроку b_i золота. Второй квест требует c_i минут и приносит d_i золота соответственно. Игрок выбирает ровно один квест из двух, выполняет его, после чего моментально встречается другого случайного персонажа, и так далее.

Павел будет играть в эту игру бесконечно долго. Насколько быстро он сможет зарабатывать золото при оптимальной игре?

Более формально, пусть t — время, проведенное Павлом в игре, а $g(t)$ — количество золота, которое сможет заработать Павел за t минут игры. Вы должны найти $\lim_{t \rightarrow \infty} \frac{g(t)}{t}$.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 200\,000$) — количество персонажей в игре.

В каждой из следующих n строк записаны четыре целых числа a_i, b_i, c_i и d_i ($1 \leq a_i, b_i, c_i, d_i \leq 10^9$) — продолжительность первого квеста, награда за первый квест, продолжительность второго квеста и награда за второй квест соответственно.

Формат выходных данных

Выведите одно число — максимальную скорость зарабатывания золота.

Абсолютная или относительная погрешность ответа не должна превышать 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
2 1 10 10 70 1 1 10 20	6.454545454545455
2 1 20 100 100 2 1 2 1	7.000000000000000

Задача Н. Средний вес дерева

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Пусть есть помеченное дерево, то есть вершины дерева пронумерованы числами от 1 до n .

Определим вес дерева так: сумма по всем ребрам (uv) величин $u \cdot sz_{uv}(u) + v \cdot sz_{vu}(v)$. Здесь $sz_{vu}(v)$ — это размер того поддерева, в котором окажется v после удаления ребра (vu) .

Дан массив a размера n . Элементы массива либо являются целыми числами от 1 до $n - 1$, либо равны -1 . v -й элемент соответствует степени вершины v . Назовём дерево размера n *хорошим*, если для всех v таких, что $a_v \neq -1$, верно, что степень вершины v равна a_v . Другими словами, если $a_v = -1$, то степень вершины v может быть любой, а иначе степень фиксирована и равна a_v .

Выберем одно *хорошее* дерево случайно и равномерно. Пусть E — матожидание веса этого дерева. Найдите целую часть E .

Формат входных данных

В первой строке записано одно число n — размер дерева ($2 \leq n \leq 10^6$).

Во второй строке записан массив длины n , все элементы которого являются либо числами от 1 до $n - 1$ (степень фиксирована), либо равны -1 (степень может быть любой). Сумма модулей элементов массива не превосходит $2n - 2$.

Формат выходных данных

Выведите целую часть E .

Примеры

стандартный ввод	стандартный вывод
5 1 -1 -1 -1 -1	67
5 -1 -1 -1 -1 1	52
4 1 1 1 3	42
4 1 1 2 2	38

Задача I. Грустные преподаватели

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В летней школе по программированию работают n преподавателей, которые никогда не унывают. Смена подходит к концу, и всем преподавателям требуется попасть к бухгалтеру на прием.

Бухгалтер работает без перерывов в промежуток времени от 0 до T минут. Для того, чтобы обслужить одного преподавателя, бухгалтеру требуется **одна** минута.

Каждый преподаватель хочет попасть к бухгалтеру на прием ровно один раз. Так как преподаватели очень занятые люди, они не знают, в какой момент времени они пойдут к бухгалтеру, поэтому каждый из них равновероятно выбирает случайный (не обязательно целый) момент времени от 0 до T и идет к бухгалтеру в этот момент времени. Преподаватели очень обидчивые, поэтому если бухгалтер не успеет обслужить какого-то преподавателя до конца рабочего дня, тот расстроится. Также, если какой-то преподаватель придет на прием в момент времени, когда бухгалтер обслуживает кого-то другого, он расстроится.

Вычислите вероятность того, что бухгалтер обслужит всех преподавателей, и никто из них не расстроится. Если искомая вероятность равна $\frac{P}{Q}$, требуется вывести величину $P \cdot Q^{-1} \bmod 998244353$. Здесь Q^{-1} — это обратный к Q элемент по модулю 998244353.

Формат входных данных

В единственной строке через пробел записаны два числа n и T ($1 \leq n, T \leq 10^8$).

Формат выходных данных

Пусть искомая вероятность равна $\frac{P}{Q}$. Выведите единственное число — $P \cdot Q^{-1} \bmod 998244353$.

Примеры

стандартный ввод	стандартный вывод
1 2	499122177
5 60	245344771
5 3	0

Задача J. Игры уголовников

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Случилось ужасное! Товарищ майор узнал, что вы репостнули мем про Иисуса, написали комментарий, оскорбляющий власть, лайкнули очередной видосик Навального, да и ещё и сходили на нанесанкционированный митинг! Теперь вы официально экстремист, а за это вам грозит тюремный срок на 100 лет!

В тюрьме вы сразу повстречали заядлого уголовника Антона, который предложил вам сыграть в особую тюремную игру уголовников. Игра заключается в следующем: есть 2 игрока, один — чётный, а другой — нечётный. В каждом раунде игры оба игрока показывают один или два пальца. Если a — сумма показанных пальцев, то если a чётно, то нечётный игрок платит чётному a монет, а если a нечётно, то чётный игрок платит нечётному a монет. Условия одинаковы, поэтому Антон предложил вам самим выбрать, чётным вы будете или нечётным. После этого каждый день своего десятилетнего заключения (всего $365 * 100 = 36500$ дней) вы сыграете с Антоном по партии. Ваша цель — не остаться в минусе и в конце своего заключения не потерять свои деньги после всех 36500 партий. Вам лениво каждый раз приходиться к Антону и играть с ним в эту игру, поэтому вы решили написать программу, играющую за вас.

Протокол взаимодействия

Это интерактивная задача.

В первой строке выведете 0, если вы хотите играть за чётного игрока, или 1, если вы хотите играть за нечётного.

После этого следующие 36500 ходов пройдут следующим образом. Выведите одно число 1 или 2 — количество пальцев, которое вы показываете в этом раунде. В ответ на это программа жюри, имитирующая Антона, выведет вам одно число 1 или 2 — количество пальцев, которое показывает Антон. Гарантируется, что в i -й ход выведенное интерактором число никак не зависит от того, какое число вы вывели в i -й ход, т.е. программа Антон играет честно.

После 36500 ходов программа жюри выведет ваш баланс — разницу полученных и отданных вами монет. В случае, если баланс неотрицательный, ваша программа будет считаться верной.

Все строки завершайте символом перевода строки и сбросом буфера ввода. Все числа программы жюри выводятся в новой строке. Для удобства в примере показана игра, состоящая из 4 партий.

Пример

стандартный ввод	стандартный вывод
	0
1	1
2	1
1	2
2	2
0	

Задача К. Игра с монеткой

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Два игрока играют в простую игру с монеткой. У них есть набор из нескольких строк одинаковой длины. Строки содержат только символы «Н» и «Т», обозначающие орел и решку.

Первый игрок выбирает одну из последовательностей из множества, затем второй игрок делает то же самое.

После этого игроки бросают честную монету, пока одна из выбранных последовательностей не появится как набор последовательных бросков. Игрок, который выбрал эту последовательность, побеждает.

Вам заданы последовательности, которые выбрали игроки. Найдите вероятность того, что первый игрок выиграет, при условии что оба игрока играют оптимально.

Формат входных данных

Первая строка содержит число n ($2 \leq n \leq 50$) — число строк в наборе. Следующие n строк содержат различные строки из символов «Н» и «Т». Длина строк одинакова и не больше 10.

Формат выходных данных

Выведите вероятность того, что выиграет первый игрок, с точностью не менее 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
2 Н Т	0.5
2 НННННННН ТТТТТТТТ	0.5
2 НН НТ	0.5
2 ТТТ НТТ	0.875
8 ННН ННТ НТН НТТ ТНН ТНТ ТТН ТТТ	0.333333

Задача L. Бешеный конь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Бешеный конь стоит на шахматной доске размера $n \times m$. Каждый раз он с вероятностью $1/8$ делает один из 8 ходов. Если в результате хода конь оказывается за пределами доски, игра для него трагически заканчивается. Для каждой клетки доски найдите математическое ожидание числа ходов, если эта клетка будет начальной.

Формат входных данных

Ввод содержит числа n и m ($n \times m \leq 100$).

Формат выходных данных

Выведите таблицу $n \times m$, в каждой ячейке которой написан ответ для соответствующей клетки. Абсолютная или относительная погрешность ответа должна быть не более 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 3	1.3333333333 1.3333333333 1.3333333333 1.3333333333 1.0000000000 1.3333333333 1.3333333333 1.3333333333 1.3333333333
5 2	1.1612903226 1.1612903226 1.1428571429 1.1428571429 1.2903225806 1.2903225806 1.1428571429 1.1428571429 1.1612903226 1.1612903226

Задача М. Неисправный робот

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Робот, вышедший из-под контроля, находится на плоскости и делает k случайных шагов. На каждом шаге робот выбирает случайно одно из четырех направлений и делает шаг в этом направлении. Вероятность того, что робот выберет шаг на север, юг, восток или запад равна n , s , e и w процентов, соответственно.

Путь робота называется простым, если он посещает каждую точку не более одного раза (стартовая точка робота это всегда первая посещенная точка). Посчитайте вероятность того, что путь робота окажется простым. Например, «EENE» и «ENW» являются простыми, а «ENWS» и «WWWWSNE» — нет («N», «S», «E» и «W» обозначают шаг на север, юг, восток и запад, соответственно). Посчитайте вероятность того, что путь робота окажется простым.

Формат входных данных

В первой строке входного файла содержатся числа k , n , s , e и w ($1 \leq k \leq 14$, $n, s, e, w \geq 0$, $n + s + e + w = 100$) — количество шагов и вероятности сделать шаг на север, юг, восток и запад в процентах.

Формат выходных данных

Выведите одно вещественное число — вероятность того, что путь робота будет простым. Абсолютная или относительная погрешность не должна превосходить 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
1 25 25 25 25	1.0
2 25 25 25 25	0.75
7 4 75 13 8	0.6757631663232001

Задача N. Эскалатор

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

Илья устал от олимпиадного программирования, ушёл из университета и устроился на работу в мажораналде метрополитен. Перед ним поставили задачу определения нагрузки на эскалатор.

Пусть n человек стоят в очереди на эскалатор. В каждую секунду происходит одно из двух: либо первый человек в очереди с вероятностью p заходит на эскалатор, либо первый человек в очереди с вероятностью $1 - p$ остаётся стоять на месте, не в силах совладать с боязнью эскалаторов, задерживая при этом всю очередь за ним.

i -й в очереди не сможет зайти на эскалатор, пока на него не зайдут люди с номерами от 1 до $i - 1$ включительно. За одну секунду может зайти только один человек. Так как эскалатор бесконечный, то, единожды зайдя на него, человек никогда с него не сойдёт, т. е. будет ехать на нем в эту и в любую последующую секунды. Илье нужно посчитать математическое ожидание количества людей, которые будут находиться на эскалаторе после t секунд.

Вам необходимо помочь ему в решении этой непростой задачи.

Формат входных данных

В единственной строке входных данных заданы три числа n, p, t ($1 \leq n, t \leq 2000, 0 \leq p \leq 1$). Числа n и t — целые, число p — вещественное, заданное ровно с двумя знаками после запятой.

Формат выходных данных

Выведите одно вещественное число — математическое ожидание количества людей, которые будут на эскалаторе через t секунд. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
1 0.50 1	0.5000000000
1 0.50 4	0.9375000000
4 0.20 2	0.4000000000

Задача О. Линейные уравнения

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned}a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ &\dots \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n\end{aligned}$$

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leq n \leq 20$). В следующих n строках записано по $n + 1$ целых чисел: $a_{i1}, \dots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать n чисел x_1, \dots, x_n , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333333333 0.666666667

Задача Р. Автоматический ответатель

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Жюри олимпиады надоело отвечать на вопросы участников, поэтому они собрали автоматический ответатель. У ответателя есть $n + 2$ состояния. Изначально он находится в состоянии 1. Состояния $n + 1$ и $n + 2$ являются терминальными. Если ответатель оказывается в состоянии $n + 1$, он отвечает «Без комментариев», если он оказывается в состоянии $n + 2$ — то «Смотри условие». Состояния от 1 до n являются промежуточными. Если ответатель находится в состоянии i , то на следующем шаге он перейдет в состояние j с вероятностью $p_{i,j}$.

Вам дано описание ответателя. Определите, с какой вероятностью он отвечает «Без комментариев».

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 50$). Следующие n строк содержат по $n + 2$ числа — вероятности $p_{i,j}$ в процентах ($1 \leq p_{i,j} \leq 99$).

Формат выходных данных

Выведите одно число — вероятность того, что ответатель ответит «Без комментариев». Относительная или абсолютная погрешность не должна превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
1 50 20 30	0.4

Задача Q. Двоичный Гаусс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из n битовых векторов размера n и еще один вектор. Получить этот вектор как хог исходных.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 300$). В следующих n строках записаны исходные вектора, в последней строке записан вектор, который нужно получить.

Формат выходных данных

Если решений нет, выведите **No solution**. Если решений несколько, выведите **Multiple solutions**. Если решение единственное, выведите номера векторов (вектора нумеруются с 0), которые нужно сложить, чтобы получить данный вектор. Номера выведите в порядке возрастания.

Примеры

стандартный ввод	стандартный вывод
3 100 111 101 010	1 2
3 100 111 011 010	No solution
3 111 010 101 000	Multiple solutions

Задача R. Добавление векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В изначально пустое множество один за другим добавляются m битовых векторов размера n . После добавления каждого скажите, можно ли его представить как хог векторов, добавленных до него.

Формат входных данных

В первой строке записаны числа n и m ($1 \leq n \leq 50$, $1 \leq m \leq 10000$). В следующих m строках записаны вектора.

Формат выходных данных

Для каждого вектора выведите 1, если его можно представить как хог предыдущих и 0 если нельзя.

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	0 0
3 4 100 111 011 010	0 0 1 0
3 4 000 111 111 111	1 0 1 1

Задача S. Максимальный хог

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из n чисел. Найдите максимальное число, которое можно получить как битовый хог данных.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 1000$). В следующих n строках записаны исходные числа a_i ($0 \leq a_i \leq 2^{63} - 1$).

Формат выходных данных

Выведите максимальное число, которые можно получить.

Примеры

стандартный ввод	стандартный вывод
2 1 2	3
3 6 2 8	14
3 15 3 8	15

Задача Т. Число возможных векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из m битовых векторов размера n ($1 \leq n, m \leq 50$). Сколько различных векторов можно получить как хог этих векторов.

Формат входных данных

В первой строке записаны числа n и m . В следующих m строках записаны вектора.

Формат выходных данных

Выведите число векторов, которые можно получить

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	4
3 4 100 111 011 010	8
3 4 111 111 111 000	2