

Задача 1. Гласность

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Посчитайте количество гласных букв в предложении текста на английском языке. Гласными буквами в английском языке являются буквы “a”, “e”, “i”, “o”, “u”, “y”.

Входные данные

Заканчивающаяся точкой последовательность символов (не более 500 символов), среди которых могут быть прописные и строчные латинские буквы, пробелы, знаки препинания (кроме точки), цифры.

Выходные данные

В первой строке одно целое число – количество гласных букв в заданной последовательности символов.

Пример:

Входные данные (одной строкой)	Выходные данные
The quick brown fox jumps over the lazy dog (thE qUIck brOwn fOx jUmps OvEr thE lAzY dOg).	24

Задача 2. Велосипедист

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Любитель велосипедных прогулок ежедневно ездит на велосипеде из дома на работу и обратно. На работу он едет со скоростью X км/ч, обратно со скоростью Y км/ч. Определите среднюю скорость велосипедиста. Средняя скорость по определению равна скорости такого равномерного движения, при котором тело прошло бы данный участок пути за тот же промежуток времени, что и при действительном движении.

Входные данные

В первой строке два целых числа через пробел – X и Y ($1 \leq X, Y \leq 100$).

Выходные данные

Средняя скорость в км/ч в виде несократимой дроби – два целых числа (числитель и знаменатель), разделенные символом “/”. Если знаменатель равен 1, то символ “/” и знаменатель не выводятся.

Пример:

Входные данные	Выходные данные
10 20	40/3

Задача 3. Азартные игры

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Один любитель азартных игр постоянно проигрывал в игровых автоматах. Он просил своего приятеля-программиста разработать ему программу для КПК, которая могла бы ему в оценке шансов в играх. Входными данными должны быть два числа: N – максимальное число ходов в игре и M – число равновероятных исходов при одном ходе (например, при $M = 6$ получаем игру с кубиком, при $M = 13$ – с волчком как в игре «Что? Где? Когда?» и т.д.). Все исходы занумерованы от 1 до M . После любого хода (вплоть до N , когда игра сама закончится) игрок может остановиться, и результатом игры в этом случае будет номер последнего исхода. Нужно определить математическое ожидание результата игры при оптимальной стратегии играющего, то есть стратегии, которая максимизирует это математическое ожидание.

Справка:

Вероятность – это отношение количества тех вариантов, при которых рассматриваемое событие наступило, к общему количеству вариантов.

Математическое ожидание случайной величины – это сумма всех возможных значений этой величины с весами, равными вероятности появления этих значений.

Входные данные

В первой строке 2 натуральных числа M и N , разделенные пробелом ($1 \leq N \leq 100$, $1 \leq M \leq 100$).

Выходные данные

В первой строке – одно вещественное число в формате с фиксированной точкой и тремя знаками после точки.

Пример:

Входные данные	Выходные данные
6 1	3.500

Задача 4. Убитые еноты

Ограничение по времени: 2 секунды

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

В одной стране правительство решили бороться с социальным неравенством довольно простым способом — каждый день за круглый стол приглашалось N человек, состояние которых в этот день планировалось уравнивать. Состояние каждого из них оценивалось некоторым целым числом условных единиц (у.е.), при этом для некоторых людей это число могло быть и отрицательным (кредитные программы в стране работали отлично). Далее правительство предлагало следующий механизм уравнивания состояний — за один шаг произвольный человек отдает по одной у.е. своим соседям, при этом еще одну у.е. он должен был отдать государству в качестве оплаты за работу правительства. Эта процедура продолжалась до тех пор, пока состояния всех участников процесса не уравнивались или не наступал конец рабочего дня.

Разумеется, в результате такого уравнивания даже те, кто изначально имел достаточно большое состояние, могли уйти в минус, но это правителей не смущало.

Когда выяснилось, что иногда уравнивать состояния такой нехитрой процедурой просто невозможно, и за день все участники могли стать нищими, народ взбунтовался, и правительству пришлось пойти на уступки — теперь они обязались гарантировать, что уравнивать состояния приглашенных людей с помощью данной процедуры реально. Для этого правителям нужна программа, которая для заданного набора людей определит, возможно ли уравнивать их состояния с помощью вышеописанной процедуры и, если возможно, ~~какую минимальную сумму правительство при этом заработает~~ за какое минимальное количество шагов это произойдет.

Входные данные:

В первой строке одно натуральное число N ($3 \leq N \leq 10000$) - количество приглашенных для уравнивания человек.

Во второй строке через пробел N целых чисел X_i ($-5000 \leq X_i \leq 5000$) - состояние каждого из приглашенных в у.е.

Выходные данные:

В первой и единственной строке одно целое число – минимальное количество шагов, за которое можно уравнивать состояния приглашенных людей. Если уравнивать нельзя, вывести «-1».

Пример:

Входные данные	Выходные данные
8 0 4 0 1 0 4 0 1	2

Задача 5. Главное - ориентация

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Задана произвольная развертка куба. На грани этой развертки нанесены цифры от 1 до 6, при этом каждая цифра имеет стандартную ориентацию на плоскости. Ориентацию грани будем считать совпадающей с ориентацией цифры. Таким образом, для каждой грани определены верхнее, правое, нижнее и левое ребра. Требуется определить, какие грани будут соседними на кубе и по какому ребру.

Входные данные:

6 строк по 6 цифр (от 0 до 6) через пробел. 0 обозначает пустое место, а цифры от 1 до 6 задают развертку куба. Гарантируется, что развертка корректна.

Выходные данные:

6 строк по 4 цифры через пробел. В i -ой строке номера граней, соседних с гранью под номером i , в следующем порядке:

- 1) грань, соседняя по верхнему ребру;
- 2) грань, соседняя по правому ребру;
- 3) грань, соседняя по нижнему ребру;
- 4) грань, соседняя по левому ребру.

Пример:

Входные данные	Выходные данные
0 0 5 0 0 0	5 2 6 4
0 1 2 3 4 0	5 3 6 1
0 0 6 0 0 0	5 4 6 2
0 0 0 0 0 0	5 1 6 3
0 0 0 0 0 0	4 3 2 1
0 0 0 0 0 0	2 3 4 1

Задача 6. Дан приказ

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Карта местности, на которой происходило сражение, имела форму квадрата со стороной, равной степени двойки, и была разбита на единичные квадраты, пронумерованные хитрым образом (чтобы команды «перевести отряд из 2 в 15» и им подобные, будучи перехваченными врагом, не раскрыли, откуда и куда так перемещается отряд). Итак, сначала пронумеровали первые четыре клетки:

1 3
2 4

Затем нумерацию продолжили:

1 3 9 11
2 4 10 12
5 7 13 15
6 8 14 16

Далее аналогичным образом получили

1 3 9 11 33 35 41 43
2 4 10 12 34 36 42 44
5 7 13 15 37 39 45 47
6 8 14 16 38 40 46 48
17 19 25 27 49 51 57 59
18 20 26 28 50 52 58 60
21 23 29 31 53 55 61 63
22 24 30 32 54 56 62 64

Таким образом нумерацию продлили на всю карту.

Вскоре выяснилось, что не только враг, но и друг затрудняется определить, какой отряд и куда должен переместиться — нумерация по номеру строк и столбцов все таки привычнее. Требуется по заданному номеру клетки определить номера строки и столбца, в которых эта клетка находится. Строки нумеруются с 1 сверху вниз. Столбцы нумеруются с 1 слева направо.

Входные данные:

В первой и единственной строке одно натуральное число K ($1 \leq K \leq 1\,000\,000\,000$) - номер клетки.

Выходные данные:

В первой и единственной строке через пробел два натуральных числа — номера строки и столбца клетки с номером K .

Пример:

Входные данные	Выходные данные
37	3 5

Задача 7. Все идет по проводам

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Территория секретной базы, имеющая форму многоугольника (возможно, невыпуклого) тщательно охранялась. В вершинах этого многоугольника, а местами и просто на его границах, стояли башни с наблюдательными постами. Связь между башнями осуществлялась через провода. Изначально, провода соединяли каждые две соседние по периметру башни, причем провод шел по прямой от одной башни к другой. Очевидно, что такая система связи была очень ненадежной, потому давно стоял вопрос о модернизации. И вот, наконец, ее время пришло.

Начальство предложило следующую идею - выбрать несколько башен и провести от каждой из них провода до всех остальных башен. К сожалению, эта идея оказалась не выполнима, так как провода протягивались опять таки только по прямой, а пускать провод вне территории базы было строго запрещено. Тогда планы слегка изменились — было решено из всех башен выбрать те, из которых можно провести максимально возможное количество проводов до других по территории базы или по ее границе. Помогите определить эти башни.

Входные данные:

В первой строке N — общее количество башен ($3 \leq N \leq 100$).

Далее N строк по 2 целых числа - координаты башен в порядке обхода по периметру территории базы, оставляющем саму территорию слева. Все координаты по модулю не превышают 100000.

Выходные данные:

В первой строке натуральное число K - количество башен, которые можно соединить проводами, идущими по территории базы или по ее границе, с максимально возможным количеством башен, и M — количество проводов, которое будет исходить из каждой такой башни.

Во второй строке K номеров выбранных башен через пробел в порядке возрастания их номеров. Башни нумеруются с 1 в порядке следования их координат во входных данных.

Пример:

Входные данные	Выходные данные
12	4 11
3 3	3 6 9 12
3 6	
0 6	
0 9	
-3 9	
-3 6	
-6 6	
-6 3	
-3 3	
-3 0	
0 0	
0 3	

Задача 8. Чертили черными чернилами чертеж

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Пусть дано бесконечное клеточное поле, все клетки которого имеют белый цвет. В одну произвольную клетку ставят робота, который закрашивает в черный цвет ту клетку, в которой он находится. Далее робот несколько раз перемещается по полю на произвольное количество клеток в одном из четырех направлений, закрашивая при этом все клетки, через которые проходит. Требуется определить общее количество закрашенных клеток после того, как робот остановится.

Входные данные:

В первой строке одно натуральное число N , $1 \leq N \leq 10\,000$ - количество перемещений робота. Далее N строк, каждая из которых содержит один символ (R, L, U, D) - направление движения робота (право, лево, вперед, назад) и через пробел число m - количество клеток, на которые робот перемещается, $1 \leq m \leq 10\,000$. При перемещении робот не поворачивается.

Выходные данные:

В первой и единственной строке одно натуральное число K — общее количество закрашенных роботом клеток.

Пример:

Входные данные	Выходные данные
6 R 1 U 1 R 1 D 1 L 3 D 1	7

Задача 9. Точки на сфере

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Начинающий программист пишет игровую программу по мотивам звездных войн. Ему нужно запрограммировать взрыв атомной бомбы. Область поражения распространяется по сфере, в каждый момент времени радиус сферы легко вычисляется. Нужно взрывать корабли в тот момент, когда их достигает область поражения. Требуется оценить, сколько кораблей максимум могут оказаться подрываемыми одновременно. Корабли могут быть расположены в прямоугольной пространственной сетке с единичным шагом по всем направлениям (естественно, радиус задается в тех же единицах). Другими словами, требуется определить, сколько точек с целочисленными координатами лежит на сфере заданного радиуса с центром в точке $(0, 0, 0)$.

Входные данные

В первой строке одно целое число R^2 ($1 \leq R^2 \leq 10\,000\,000$) – квадрат радиуса сферы – границы области поражения в текущий момент времени.

Выходные данные

В первой и единственной строке одно целое число – максимально возможное число подрываемых одновременно кораблей.

Пример:

Входные данные	Выходные данные
1	6

Задача 10. Карточка

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

В библиотеке университета делали ремонт. После ремонта выяснилось, что в карточке пропало M карточек. Все карточки были пронумерованы с 1 до N , электронный каталог сохранился. Поэтому для восстановления работоспособности картотеки требуется заново распечатать отсутствующие карточки. Определите номера карточек, отсутствующие среди номеров, представленных в списке.

Входные данные

В первой строке два положительных целых числа N и M через пробел ($0 < M < N \leq 10000$). Далее $N-M$ строк, в каждой строке одно число – номер карточки.

Выходные данные

M строк, в каждой строке одно целое число – номер потерянной карточки. Номера потерянных карточек упорядочить по возрастанию.

Пример:

Входные данные	Выходные данные
10 5	2
1	4
3	5
7	8
6	10
9	

Задача 11. Тарелка

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

В один из первых дней медового месяца молодой муж мыл посуду и разбил тарелку. Он собрал кусочки тарелки, но не уверен, что все. Помогите молодожену избежать скандала и определить, все ли куски тарелки собраны. Тарелка была плоская, круглая, радиуса R . Все куски являются «многоугольниками» (всегда выпуклыми), у которых не более одного участка границы может быть дугой окружности радиуса R (причем длина этой дуги всегда меньше половины длины целой окружности), а остальные — отрезками прямой. Известно, что среди собранных кусков нет лишних (других тарелок молодожены не разбивали).

Входные данные

В первой строке одно вещественное число R ($0.1 \leq R \leq 10$) — радиус разбитой тарелки. В следующей строке одно целое положительное число n , $1 \leq n \leq 1000$, где n — число кусков, которые собрал молодой муж. Далее следует n описаний кусков. Описание одного куска в первой строке содержит число фрагментов границы k , $2 \leq k \leq 1000$ и, через пробел, признак того, что у этого куска есть фрагмент — дуга (1 — есть дуга, 0 — нет дуги). Далее $k-1$ строк, в каждой из которых описание одного фрагмента границы — 2 вещественных числа: две координаты точки конца фрагмента (каждый фрагмент начинается там, где кончается предыдущий, начало первого фрагмента и конец последнего всегда в точке $(0, 0)$, если в куске есть фрагмент-дуга, то он всегда первый). Общее число строк в файле не превышает 10000. Все вещественные числа. Все вещественные числа представлены в формате с фиксированной точкой с точностью не более 6 знаков после запятой. При этом незначащие нули в конце числа, а в случае целых чисел и точка, могут отсутствовать.

Выходные данные

Слово «YES» в первой строке, если тарелка собрана полностью (т.е. если суммарная площадь всех потерянных кусков не превышает 0.1% от площади тарелки), «NO» в противном случае. Заметим, что из-за погрешности оцифровки вершин кусков тарелки суммарная площадь всех найденных кусков может оказаться даже больше площади тарелки, в этом случае считать, что тарелка собрана полностью

Пример 1:

Входные данные	Выходные данные
2	NO
4	
3 0	
-2 -2	
0 -2	
3 1	
2 -2	
2 0	
3 1	
2 2	
0 2	
3 1	
-2 2	
-2 0	

Пример 2:

Входные данные	Выходные данные
2	YES
5	
3 0	
-2 -2	
0 -2	
3 1	
2 -2	
2 0	
3 1	
2 2	
0 2	
3 1	
-2 2	
-2 0	
2 1	
-2 2	

Задача 12. Дорогой длинною*Ограничение по времени: 5 секунд**Ограничение по памяти: 256Мб**Входные данные: стандартный ввод**Выходные данные: стандартный вывод*

На бесконечном клеточном поле заданы K отрезков, проходящих строго по границам клеток, причем концы этих отрезков лежат в углах клеток. Требуется определить, можно ли из одного угла клетки дойти до другого, перемещаясь по этим отрезкам. Переходить с одного отрезка на другой возможно в том случае, когда эти отрезки пересекаются, т.е. имеют хотя бы одну общую точку.

Входные данные:

В первой строке одно натуральное число K ($3 \leq K \leq 20000$) - количество отрезков. Далее K строк по 4 целых числа через пробел: $X1, Y1, X2, Y2$ - координаты начала и конца каждого из отрезков.

Далее еще одна строка, содержащая 4 целых числа через пробел - XS, YS, XE, YE - координаты начальной и конечной точек.

Все координаты по модулю не превышают 10000.

Выходные данные:

Если существует путь из начальной точки в конечную, проходящий по заданным отрезкам, вывести «YES». Иначе вывести «NO».

Пример 1:

Входные данные	Выходные данные
3 0 0 2 0 0 4 0 2 0 2 2 0 0 2 2	YES

Пример 2:

Входные данные	Выходные данные
3 0 0 2 0 0 4 0 2 1 2 2 0 0 2 2	NO