

Задача 1. Гирлянда

Входные данные: *стандартный ввод*

Выходные данные: *стандартный вывод*

Ограничение по времени: *1 секунда*

Ограничение по памяти: *256Мб*

Есть N лампочек, для каждой задан период времени, через который лампочка вспыхивает, в секундах. В начальный момент времени все лампочки вспыхнули. Определите, когда они вспыхнут одновременно в следующий раз.

Входные данные:

В первой строке одно натуральное число N – количество лампочек ($2 \leq N \leq 10$).

Во второй строке N натуральных чисел через пробел – периоды вспыхивания лампочек, каждое из этих чисел не превышает 100.

Выходные данные:

В единственной строке одно значение – время в секундах, через которое все лампочки вспыхнут одновременно.

Пример:

Ввод	Вывод
3 3 6 8	24

Задача 2. Ванна

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Молодожены-программисты пришли в магазин выбирать ванну и обнаружили, что из-за сложной формы дна ванн непонятно, на какую высоту поднимается вода в каждой их них, если наливать одинаковый объем воды. Тогда они сфотографировали ванны с разных ракурсов и дома занялись компьютерным моделированием. Достаточно скоро они построили триангуляции поверхностей ванн, осталось только рассчитать высоту воды при заданном объеме. Все построенные триангуляции поверхностей являются односвязными и содержат только такие треугольники, проекции которых на плоскость XU не пересекаются и не вырождаются в отрезок. Кроме того, дно ванны не содержит локальных максимумов и горизонтальные участки могут быть только на максимальной глубине. Для упрощения описания ванн молодожены не стали задавать внешние вертикальные границы, считая, что заданная поверхность окружена вертикальными стенками, верхняя кромка которых совпадает с нулевой глубиной.

Входные данные

В первой строке два натуральных числа, разделенных не менее чем одним пробелом, n ($1 \leq n \leq 200$) – число вершин триангуляции и m ($1 \leq m \leq 400$) – число треугольников. В каждой из следующих n строк заданы вершины триангуляции, по 3 вещественных числа в строке, координаты в метрах, диапазон значений каждой координаты от 0 до 10, третья координата означает глубину в точке. В каждой из следующих m строк заданы треугольники, по 3 натуральных числа в строке, каждое число – порядковый номер вершины триангуляции. В последней строке одно натуральное число – объем наливаемой воды, измеряемый в литрах. Вода всегда наливается так, чтобы высота подъема не превышала максимальной глубины ванны.

Выходные данные

В единственной строке одно вещественное число, округленное до трех знаков после десятичной точки, в формате с фиксированной точкой – высота подъема воды относительно самой глубокой точки ванны.

Пример 1 (кубическая ванна):

Ввод	Вывод
4 2 0 0 1 1 0 1 0 1 1 1 1 1 1 2 3 2 3 4 10	0.010

Пример 2 (пирамида с треугольным основанием):

Ввод	Вывод
4 3 0 0 0 1 0 0 0 1 0 0.2 0.2 1 1 2 4 2 3 4 1 3 4 10	0.391

Задача 3. Парковка

Входные данные: *стандартный ввод*

Выходные данные: *стандартный вывод*

Ограничение по времени: *1 секунда*

Ограничение по памяти: *256Мб*

Главврач больницы простыл и не мог оперировать. Он сидел в своем кабинете и смотрел в окно, как приезжают и уезжают машины. Тут подъехала скорая, не смогла припарковаться у входа и санитарам пришлось нести больного на носилках через весь двор. “Бардак!” - подумал главврач, вызвал штатного программиста и попросил провести исследование процессов парковки.

Пусть место для парковки разбито по длине линиями разметки на L равных секций, а каждой машине скорой помощи для парковки требуется ровно D последовательных секций. При этом выбор места для парковки из всех возможных вариантов осуществляется равновероятно. Водители скорой помощи мастера парковки, поэтому могут припарковаться, даже если свободное место в точности равно D . Новые автомобили приезжают постоянно, пока есть возможность припарковаться. Требуется определить ожидаемое количество припарковавшихся автомобилей, чтобы затем выяснить, насколько необходимо расширять парковку.

Входные данные

В первой строке два натуральных числа L и D , разделенных пробелом, $3 \leq L \leq 50$ и $3 \leq D \leq L$.

Выходные данные

В единственной строке два целых числа, разделенных знаком “/”, без пробелов – математическое ожидание числа припарковавшихся автомобилей в виде несократимой дроби (даже если знаменатель дроби равен 1, он все равно выводится).

Пример:

Ввод	Вывод
6 3	3/2

Задача 4. Авиапроездной

Входные данные: стандартный вывод

Выходные данные: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Одна авиакомпания проводит интересную акцию. Она продает проездные, которые действуют по следующим правилам:

- каждому авиарейсу компании присваивается некоторая стоимость в условных единицах
- проездной имеет определенный номинал (в тех же условных единицах);
- после каждого перелета с проездного списывается количество условных единиц, равное стоимости этого перелета;
- каждый следующий перелет должен начинаться в том городе, в котором закончился предыдущий;
- если оставшееся на проездном количество условных единиц недостаточно для совершения следующего перелета, то они сгорают.

Зная номинал проездного и стоимости всех авиарейсов, требуется определить, можно ли составить план перелетов так, чтобы использовать проездной полностью, не потеряв ни одной условной единицы.

Входные данные

В первой строке три натуральных числа, разделенных пробелом:

N ($1 \leq N \leq 1000$) – количество городов, в которые летают рейсы авиакомпании,

M ($1 \leq M \leq 1000$) – количество рейсов,

v – номинал проездного ($1 \leq v \leq 1000$).

В каждой из следующих M строк заданы стоимости перелетов. Первые два числа в строке v_1 и v_2 ($1 \leq v_1, v_2 \leq N$), – номера городов, связанных авиарейсом, а третье w ($1 \leq w \leq 1000$), – стоимость перелета из города v_1 в город v_2 (естественно, v_1 и v_2 различны). Числа в строке разделены пробелом.

Известно, что любые два города связаны напрямую не более чем одним авиарейсом в каждом из двух направлений.

Выходные данные

В единственной строке выходного файла нужно вывести YES, если возможно составить план перелетов, позволяющий использовать проездной полностью, в противном случае – NO.

Пример 1:

Ввод	Вывод
4 5 8 1 2 2 1 3 3 3 4 2 3 2 2 2 1 2	YES

Пример 2:

Ввод	Вывод
3 3 6 1 2 2 2 3 3 3 1 5	NO

Задача 5. Кубичность

Входные данные: *стандартный ввод*

Выходные данные: *стандартный вывод*

Ограничение по времени: *1 секунда*

Ограничение по памяти: *256Мб*

Исследовательский космический спутник в ходе изучения некоторой функциональной зависимости произвел N измерений. Чтобы передать результаты эксперимента на Землю в наиболее сжатом виде, требуется разбить всю область значений аргумента на минимальное количество отрезков с соблюдением следующего условия: на каждом из этих отрезков должен существовать полином не более чем третьей степени, совпадающий с результатами измерений во всех точках, принадлежащих этому отрезку. Также необходимо, чтобы границы получившихся отрезков являлись точками исходных измерений.

Входные данные

В первой строке одно натуральное число N ($2 \leq N \leq 100$) - количество измерений. В следующей строке N пар целых чисел, каждое из которых по модулю не превышает 1000, – результаты проведенных измерений. В каждой такой паре первое число – значение аргумента, а второе – соответствующее значение функции в этой точке. Пары перечислены в порядке возрастания координаты, расстояние между точками измерений не превышает 100. Числа в строке разделены не менее чем одним пробелом.

Выходные данные

В единственной строке одно натуральное число – минимально возможное число отрезков.

Пример:

Ввод	Вывод
10 0 0 1 1 2 2 3 3 4 4 5 4 6 4 7 4 8 4 9 4	2

Задача 6. Отрезки

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Ограничение по времени: 3 секунды

Ограничение по памяти: 256Мб

Бродя по просторам интернета, среди прочих баннеров, предлагающих решить достаточно простые задачки и получить приз, Михалыч увидел головоломку, которая почему-то привлекла его внимание. Суть головоломки заключалась в том, чтобы из имеющихся отрезков составить выпуклый многоугольник. Дополнительная сложность заключалась в том, что с помощью мыши отрезки можно было только перемещать в произвольном направлении, а вот повернуть их никак не удавалось (возможно, из-за непродуманного интерфейса). Через полчаса безуспешных попыток расставить отрезки Михалыч вдруг обнаружил на экране монитора множество открытых веб-страниц с самым разнообразным содержанием и от обиды написал программу, которая очень быстро определяет, возможно ли решить такую головоломку. Ваша задача – написать аналогичную программу на случай, если и Вам встретится такой баннер.

Входные данные

В первой строке одно натуральное число N – количество отрезков ($3 \leq N \leq 50$). Далее N строк по два целых числа X и Y – координаты концов отрезков ($-100 \leq X, Y \leq 100$). Началом всех отрезков считается начало координат.

Выходные данные

В единственной строке выходного файла нужно вывести YES, если возможно составить выпуклый многоугольник, граница которого составлена из всех имеющихся отрезков, в противном случае – NO.

Пример 1:

Ввод	Вывод
4 1 3 2 -2 0 -1 -3 0	YES

Пример 2:

Ввод	Вывод
3 1 2 4 1 2 1	NO

Задача 7. Парад

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

В спортивном параде должны были идти две колонны спортсменов с флагами, но когда колонны уже выстроились перед выходом, оказалось, что ширины дорожки для двух колонн не хватит. Экстренно было принято решение совместить две колонны в одну. Сначала хотели пустить вперед первую колонну, а следом за ней вторую, но, посоветовавшись, чтобы никого не обидеть, решили перемешать две колонны. При этом, чтобы улучшить эстетическую составляющую зрелища, чередование участников колонн должно было происходить так, чтобы сумма разностей высот соседних флагов была минимальна.

Зная высоты флагов и их очередность в каждой колонне по отдельности, определите какую минимальную сумму разностей высот соседних флагов могут получить организаторы шествия, объединив колонны, но не меняя порядок выхода участников каждой из них.

Входные данные

В первой строке два числа L_1, L_2 , задающие количество участников каждой из колонн ($1 \leq L_1, L_2 \leq 1000$). Во второй строке через пробел перечислены L_1 высот флагов первой колонны в порядке их выхода на парад. В третьей строке аналогично L_2 высот флагов второй колонны. Значения высот – целые числа от 0 до 10000.

Выходные данные

В единственной строке выходного файла вывести минимально возможное значение суммы модулей разности высот соседних флагов объединенной колонны.

Пример:

Ввод	Вывод
3 3 1 5 3 2 2 6	8

Задача 8. Гиперия

Входные данные: *стандартный ввод*

Выходные данные: *стандартный вывод*

Ограничение по времени: *1 секунда*

Ограничение по памяти: *256Мб*

Гипероны, жители далекой-далекой вселенной, решили колонизировать новую планету. Планеты в их вселенной имеют вид N -мерного куба. Гипероны решили выяснить, как лучше всего располагать города, чтобы их было максимально возможное количество на планете. В традициях гиперонов обязательно строить города на планетах в центрах k -мерных граней (вершины являются 0-мерными гранями, ребра – 1-мерными, плоские квадраты – 2-мерными и т.д.). Причем все города строятся на гранях одинаковой размерности. Но вот проблема – гипероны не знают, граней какой размерности больше всего!

Входные данные

В первой строке одно натуральное число N ($1 \leq N \leq 10^9$) – размерность куба.

Выходные данные

В единственной строке допустимые размерности граней для построения наибольшего числа городов в порядке возрастания через пробел.

Пример 1:

Ввод	Вывод
2	0 1

Пример 2:

Ввод	Вывод
3	1

Задача 9. Как украсть миллион

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Однажды Михалыч посмотрел фильм “Как украсть миллион” и узнал, что по нему собираются делать ремейк. Ему очень запомнилась сцена, в которой актер Питер О’Тул кидает бумеранг, чтобы сработала сигнализация. Он решил помочь режиссеру ремейка сделать качественные спецэффекты (иначе зачем вообще нужен ремейк замечательного фильма?) и написал программу, рассчитывающую пересечение траектории полета бумеранга с системой лучей сигнализации. Система лучей сигнализации является замкнутой ломанной линией (возможно, самопересекающейся).

Входные данные

В первой строке три целых числа X , Y и R , описывающих координаты центра окружности, являющейся траекторией полета бумеранга, и её радиус. Числа разделены не менее чем одним пробелом. Все эти числа не превышают по модулю 20000, радиус положительный.

Во второй строке одно натуральное число N ($2 \leq N \leq 1000$) – число устройств сигнализации (точек, в которых начинаются и заканчиваются лучи).

Далее в N строках по два целых числа, разделенных не менее чем одним пробелом, числа не превышают по модулю 20000 – координаты устройств сигнализации в порядке соединения их лучами, последнее соединено лучом с первым.

Выходные данные

В единственной строке YES, если бумеранг коснулся или пересек хотя бы один луч (или попал в устройство), и NO в противном случае.

Пример 1:

Ввод	Вывод
0 0 3 2 -1 4 1 4	NO

Пример 2:

Ввод	Вывод
0 0 3 2 -3 0 -3 -2	YES

Задача 10. Клавиатура

Входные данные: *стандартный ввод*

Выходные данные: *стандартный вывод*

Ограничение по времени: *1 секунда*

Ограничение по памяти: *256Мб*

Молодая инновационная фирма разработала новую клавиатуру с эргономичным дизайном. В принципе, она могла бы быть совместима с ОС Windows, поскольку у неё есть клавиши Ctrl, Win, Alt, ScrLk, NumLock, CapsLock, Left Shift, Right Shift и др. Вот только информацию о нажатости этих клавиш клавиатура передает в виде одного целого числа в десятичной системе счисления. Напишите программу, которая по заданному номеру клавиши определяет, нажата ли она, если нажатая клавиша кодируется единичным значением бита с соответствующим номером в двоичном представлении числа, биты нумеруются справа налево, начиная с нулевого.

Входные данные

В первой строке два целых числа N – код, полученный с клавиатуры ($0 \leq N < 1024$), и, через не менее чем 1 пробел, M – номер проверяемой клавиши ($0 \leq M < 10$)

Выходные данные

В единственной строке YES, если клавиша нажата, и NO в противном случае.

Пример 1:

Ввод	Вывод
127 3	YES

Пример 2:

Ввод	Вывод
128 3	NO

Задача 11. Файндворд

Входные данные: стандартный ввод

Выходные данные: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256Мб

Одним из литературных развлечений является файндворд – текст, в котором на стыке двух слов могут быть спрятаны другие слова. Например, в тексте “Был месяц *май*. *Кактус* стоял на окне.” спрятано слово “*майка*”. Найдите все спрятанные в тексте слова из заданного списка, если разделителями слов текста являются пробелы, концы строк и заданный набор знаков препинания.

Входные данные:

В первой строке три целых числа через не менее чем 1 пробел: K – число строк текста, ($1 \leq K \leq 100$), каждая строка не более 127 символов, N – число слов, которые надо найти, ($1 \leq N \leq 100$), и M – число знаков препинания ($0 \leq M \leq 10$), которые могут разделять слова. Следующие K строк содержат текст (состоящий из строчных и прописных букв латинского алфавита, цифр, пробелов и знаков препинания из заданного набора). Затем в N строках заданы искомые слова (из того же набора символов, кроме пробелов и знаков препинания, длина каждой строки не более 20 символов, все слова списка различны). Последняя строка состоит из M символов, которые следует считать знаками препинания (знаки препинания в тесте могут как отделяться пробелами, так и не отделяться).

Выходные данные:

С первой строки, по одному слову в строке найденные слова в том порядке, в каком они спрятаны в тексте (слово может быть спрятано только на стыке двух слов, а не внутри другого слова). Порядок слова определяется по его первой букве. Если спрятанное слово встречается не один раз, то его следует выводить только один раз, считая его порядком то место текста, где оно спрятано впервые. Если в одном и том же месте спрятано 2 слова, то выводить их следует в том порядке, в каком они были во входном списке. Известно, что хотя бы одно из искомым слов действительно спрятано.

Пример:

Ввод	Вывод
3 4 2 abshykcd, bbb,/Abb anbshy kjdj jjdjdd wiueqiqqwfnс kllkkl shyk bbAbb uiuiui kjdj ,/	bbAbb shyk