

III открытая олимпиада
Очный тур
Задачи

А. Волшебный лес

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайта
 Ввод: Стандартный поток ввода (stdin)
 Вывод: Стандартный поток вывода (stdout)

Гномы клана Длиннобородых очень гордятся тем, что живут около Волшебного леса, и стараются хотя бы раз в год посещать это священное место. Но попасть в Волшебный лес не так-то просто, потому что он защищён древним заклинанием. Есть только один способ преодолеть силу заклинания. Для этого нужно на рассвете вчетвером прийти к окраине леса и закопать около тысячелетнего дуба горсть золотых монет. Причем монет должно быть ровно k штук. Но и это еще не всё. Если хотя бы один из гномов принёс в лес не все свои монеты, а среди гномов такое случается нередко, то заклинание не будет разрушено.

Каждый год в селении гномов происходят поиски тех четверых гномов, которые смогут в очередной раз пройти в лес. Из соображений экономии в лес отправляется только одна четверка гномов. Проблема состоит в том, что ни один гном никогда в жизни не поделится с другим своими монетами. Поэтому нужно найти такую четвёрку гномов, чтобы у них в сумме было ровно k золотых монет.

Помогите гномам определить, сколько есть различных вариантов четверок гномов, которые смогут пройти в Волшебный лес в этом году.

Исходные данные

В первой строке записано число k – количество золотых монет, которое требуется для снятия заклинания, $1 \leq k \leq 100$. Во второй строке записано число n – количество гномов, проживающих в селении, $4 \leq n \leq 10^6$. В следующих n строках записаны числа m_1, m_2, \dots, m_n – количества золотых монет, которые есть у гномов из селения, $0 \leq m_i \leq 10^6, 1 \leq i \leq n$.

Результат

Ваша программа должна вывести одно число – количество различных четверок гномов, которые смогут пройти в Волшебный лес. Гарантируется, что это число не превышает $4 \cdot 10^{18}$.

Пример

Исходные данные	Результат
4 4 0 1 2 3	0
4 4 1 1 1 1	1
4 5 1 1 1 1 1	5

В. Игра с волчком

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайта
 Ввод: Стандартный поток ввода (*stdin*)
 Вывод: Стандартный поток вывода (*stdout*)

По кругу стоят N человек с номерами от 1 до N в порядке возрастания (игрок с номером N и игрок с номером 1 стоят рядом). Они играют в игру по следующим правилам. Каждый игрок получает стартовое количество очков, равное его номеру. Далее в каждом раунде игры судья вращает волчок. Обозначим через t количество очков игрока, на которого указал волчок. Этот игрок должен сделать следующие действия:

1. Отсчитать t человек, которые стоят вслед за ним по кругу;
2. Вычислить среднее арифметическое очков этих игроков;
3. Округлить полученное число до целого путем отбрасывания дробной части;
4. Полученное число становится новым количеством очков этого игрока.

Известно, что за время игры судья вращал волчок K раз. А также известно, на каких игроках указывал волчок на каждом из ходов. Требуется определить сумму очков всех игроков в конце игры.

Исходные данные

В первой строке записано число N – количество игроков, $2 \leq N \leq 10^6$. Во второй строке записано число K – количество раундов в игре, $0 \leq K \leq 10^6$. В следующих K строках записаны числа n_1, n_2, \dots, n_K – номера игроков, на которых указывал волчок в раундах $1, 2, \dots, K$, соответственно, $1 \leq n_i \leq N, 1 \leq i \leq K$.

Результат

Ваша программа должна вывести одно число – сумму очков всех игроков в конце игры.

Пример

Исходные данные	Результат
2 0	3
2 1 1	4
2 1 2	2

С. Забор

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайта
 Ввод: Стандартный поток ввода (stdin)
 Вывод: Стандартный поток вывода (stdout)

Несмотря на планомерное развитие сельского хозяйства в масштабах страны, огороды не теряют популярности: ещё бы, ведь происхождение выращенных на огороде овощей не вызывает сомнений.

Одна из самых важных частей огорода – ограда. Чем она выше, тем надежнее сокрыты от посторонних взглядов растения, выращиваемые на огороженной земле.

Дима недавно обзавелся своим огородом. Но вот ограды к нему почему-то не прилагалось. У Димы есть n металлических прутьев известных длин l_i , $1 \leq i \leq n$, которые он хочет использовать в качестве опорных столбов для будущей ограды. Для установки ограды требуется ровно k прутьев одинаковой длины, причём длина прутьев непременно должна быть целым числом.

Дима может разрезать прутья специальной пилой, превращая таким образом один прут в два. Теперь он хочет получить k прутьев одинаковой длины, сделав как можно меньше разрезов. При этом эти k прутьев должны быть не короче каждого из оставшихся.

Исходные данные

В первой строке через пробел даны целые числа n и k , $1 \leq n \leq 10^5$, $1 \leq k \leq 10^5$. Во второй строке через пробел даны целые числа l_i – длины прутьев, $1 \leq l_i \leq 10^4$, $1 \leq i \leq n$.

Результат

Выведите единственное число – необходимое количество разрезов. Если же разрезать прутья требуемым образом невозможно, нужно вывести строку “impossible”.

Пример

Исходные данные	Результат
4 2 1 2 10 5	1
4 3 1 2 10 5	1
1 11 10	impossible

D. Карты и булавки

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайта
 Ввод: Стандартный поток ввода (stdin)
 Вывод: Стандартный поток вывода (stdout)

На плоском столе лежит карта некоторой местности. Карта представляет собой прямоугольник $A_0B_0C_0D_0$:

- A_0 : левый нижний угол карты, с координатами $(0, 0)$;
- B_0 : левый верхний угол карты, с координатами $(0, H)$;
- C_0 : правый верхний угол карты, с координатами (W, H) ;
- D_0 : правый нижний угол карты, с координатами $(W, 0)$.

Также на столе лежит уменьшенная копия той же самой карты местности. Уменьшенная копия карты – прямоугольник $ABCD$, точке A_0 большой карты соответствует точка A уменьшенной карты, точке B_0 – точка B и т.д. Вам требуется найти такую точку стола, в которой можно одной булавкой вертикально проткнуть обе карты таким образом, чтобы образованные “дырки” на обоих картах соответствовали одной и той же географической точке.

Исходные данные

В первой строке заданы два вещественных числа W и H – ширина и высота большой карты, соответственно, $1.0 \leq W, H \leq 1000.0$. В следующих четырех строках заданы координаты точек A, B, C и D . Каждая точка задается парой вещественных чисел $x, y, -2000.0 \leq x, y \leq 2000.0$. Все вещественные числа задаются с шестью знаками после десятичной точки. Гарантируется, что ширина уменьшенной карты не больше, чем $0.9 \cdot W$ и не меньше, чем $0.01 \cdot W$. Можно считать, что у карт нет “рамок”, т.е. любая точка прямоугольника (даже лежащая на границе) соответствует какой-то географической точке. Карту можно “протыкать” на границе.

Результат

В единственной строке выведите координаты точки, в которой нужно проткнуть обе карты булавкой. Координаты следует выводить в таком же формате, что и во вводе: через пробел, с шестью знаками после десятичной точки. Если решения задачи не существует, выведите единственное слово “NONE” (без кавычек).

Пример

Исходные данные	Результат
4.000000 2.000000 1.000000 0.000000 0.000000 0.000000 0.000000 2.000000 1.000000 2.000000	0.800000 0.400000
4.000000 2.000000 3.000000 1.000000 3.000000 2.000000 5.000000 2.000000 5.000000 1.000000	NONE

Е. Гусеница

Ограничение времени: 1 секунда
Ограничение памяти: 64 Мегабайта
Ввод: Стандартный поток ввода (*stdin*)
Вывод: Стандартный поток вывода (*stdout*)

Любимая игрушка маленькой Кати – деревянная гусеница. У этой гусеницы есть голова и n сегментов туловища. Все сегменты раскрашены по-разному, и их можно в любом порядке прикреплять к голове. Кате очень любопытно, сколько есть различных вариантов собрать гусеницу, если использовать голову и все ее n сегментов. Гусеница, естественно, должна начинаться с головы. Несмотря на то, что Катя еще маленькая и не может посчитать количество вариантов, она понимает, что их может быть очень много. Поэтому она хочет узнать только последние 20 цифр числа способов собрать гусеницу.

Помогите Кате ответить на ее вопрос.

Исходные данные

Единственная строка содержит целое число n – количество сегментов гусеницы $1 \leq n \leq 10^{18}$.

Результат

Ваша программа должна вывести последние 20 цифр числа способов собрать гусеницу. Если число способов собрать гусеницу имеет не более чем 20 цифр в своей десятичной записи, то необходимо вывести все это число.

Пример

Исходные данные	Результат
2	2

Г. Офис для программистов

Ограничение времени: 0.5 секунды
Ограничение памяти: 64 Мегабайт
Ввод: Стандартный поток ввода (*stdin*)
Вывод: Стандартный поток вывода (*stdout*)

Компания “*Word Consulting Inc.*” решила основательно расширить свой программистский департамент и построила для его размещения дом в форме шара. Бригада, нанятая для отделки помещений, умеет строить только плоские стены, каждая из которых позволяет разделить здание на две части. Смета, утвержденная для отделки, позволяет построить n таких стен ($0 \leq n \leq 10^6$). Стены могут пересекаться между собой.

Стоит отметить, что программисты предпочитают ютиться в маленьких, но индивидуальных помещениях, а не в “open space”. Сколько же программистов сможет разместить компания в этом здании, когда строительство будет завершено?

Исходные данные

В единственной строке записано неотрицательное целое число n .

Результат

Выведите единственное число – максимальное число частей, на которое может быть разбит шар заданным числом плоскостей.

Пример

Исходные данные	Результат
5	26
1000	166667501

Г. Космические пузыри

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайта
 Ввод: Стандартный поток ввода (stdin)
 Вывод: Стандартный поток вывода (stdout)

Космические пузыри – очень редкое явление, но необычайно красивое. Особенно редко случается одновременное рождение нескольких пузырей. Науке известен случай одновременного рождения сразу n космических пузырей. Пузыри в момент рождения представляют из себя сферу радиуса 0. Сразу же после рождения радиус пузыря начинает равномерно увеличиваться, при этом его центр остается неподвижен. Скорость роста пузыря не меняется до тех пор, пока он не столкнется с другим пузырьком. В момент столкновения рост пузыря прекращается, и он застывает в этом положении навсегда. Скорости роста пузырей могут отличаться.

Астрономы ломают голову, чему же будет равна сумма радиусов пузырей в тот момент, когда рост всех пузырей прекратится. Помогите астрономам найти ответ на этот вопрос. Известно, что эти n космических пузырей родились в открытом космосе вдали от других объектов, и рост каждого из них остановится от столкновений друг с другом.

Исходные данные

В первой строке содержится целое число n – количество космических пузырей $2 \leq n \leq 1000$. В следующих n строках содержится информация о пузырях. В $(i + 1)$ -ой строке записаны четыре числа x_i, y_i, z_i, v_i – координаты центра и скорость роста i -го пузыря, $-10^6 \leq x_i, y_i, z_i \leq 10^6, 1 \leq v_i \leq 10^6, 1 \leq i \leq n$.

Результат

Ваша программа должна вывести единственное число – сумму радиусов пузырей в тот момент, когда рост всех пузырей прекратится. Результат необходимо вывести с четырьмя знаками после запятой.

Пример

Исходные данные	Результат
2 0 0 0 1 2 0 0 1	2.0000
3 0 0 0 1 2 0 0 1 0 0 100 5	101.0000

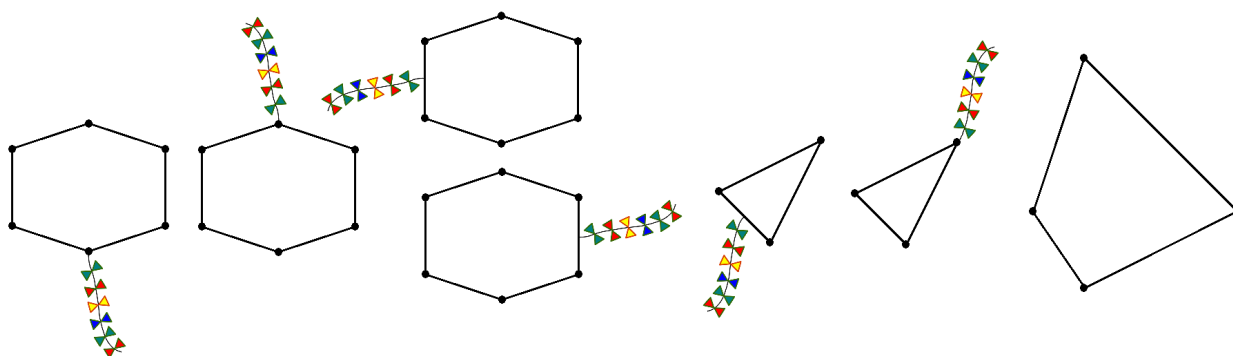
Н. Оси симметрии

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайт
 Ввод: Стандартный поток ввода (stdin)
 Вывод: Стандартный поток вывода (stdout)

Друзья привезли из путешествия по Китаю Вова в подарок набор для изготовления воздушного змея. На следующий день выдалась хорошая погода, и он решил склеить змея и запустить его. Инструкция по изготовлению, конечно, была только на китайском языке, поэтому Вова решил, что разберётся и без неё. Немного повозившись, он соорудил змея в форме плоского многоугольника, оставалось только приклеить к нему хвост.

И тут Вова пришлось задуматься над тем, к какой точке границы многоугольника должен быть приклеен хвост змея. Интуиция подсказала ему, что для того, чтобы полёт змея был устойчивым, его хвост должен лежать на некоторой оси симметрии многоугольника.

На рисунках ниже изображены два воздушных змея, летающих устойчиво и все варианты прикрепления к ним хвоста, а также один воздушный змей, летающий неустойчиво.



Сколько существует точек на границе многоугольника, таких что, если приклеить к ним хвост, в результате получится воздушный змей, летающий устойчиво?

Исходные данные

В первой строке записано единственное целое число N ($3 \leq N \leq 1000$). В следующих N строках записаны через пробел по два целых числа, не превосходящие по модулю 10000 – координаты вершин многоугольника в порядке обхода.

Результат

Вывести единственное число – количество точек на границе многоугольника, к которым можно приклеить хвост змея.

Пример

Исходные данные	Результат
6 3 3 6 2 6 -1 3 -2 0 -1 0 2	4
3 -2 2 2 4 0 0	2
4 0 6 6 0 0 -3 -2 0	0

I. Зеркало в коридоре

Ограничение времени: 0.5 секунды
 Ограничение памяти: 64 Мегабайт
 Ввод: Стандартный поток ввода (stdin)
 Вывод: Стандартный поток вывода (stdout)

Простуженный Петя лежал в постели, как вдруг кто-то открыл дверь. Пете было лень вставать и он посмотрел в зеркало, чтобы увидеть кто пришёл. Известны координаты вошедшего (его можно считать материальной точкой), необходимо узнать, удастся ли Пете увидеть отражение вошедшего в зеркале. Зеркало имеет форму круга с центром в начале координат и расположено в плоскости $ax + by + cz = 0$. Петю тоже можно считать материальной точкой.

Гарантируется, что Петя и незнакомец не лежат в плоскости зеркала и что Петя всегда видит отражающую сторону зеркала. Если изображение попадает на границу зеркала, то Петя видит вошедшего. Так как и Петю и вошедшего можно считать материальными точками, Петя может увидеть отражение вошедшего как сквозь вошедшего, так и сквозь свое собственное отражение.

Исходные данные

В первой строке записано натуральное число n – количество тестов, не превышающее 500. Каждый тест состоит из четырёх строк. В первой из них записан радиус зеркала. Во второй – коэффициенты a, b, c . Далее следуют координаты Пети. В четвёртой строке каждого теста записаны координаты вошедшего. Все числа целые и по модулю не превосходят 1000.

Результат

Выведите n строк. В каждой строке – ответ на очередной тест. “Yes”, если Петя видит вошедшего и “No” в противном случае.

Пример

Исходные данные	Результат
2	Yes
2	No
0 1 0	
0 2 0	
0 1 0	
1	
0 1 0	
2 -2 0	
1 1 0	

J. Конструктор “ЛЕВО”

Ограничение времени: 1 секунда
 Ограничение памяти: 64 Мегабайта
 Ввод: Стандартный поток ввода (*stdin*)
 Вывод: Стандартный поток вывода (*stdout*)

Конструктор “ЛЕВО” состоит из n одинаковых деталей. Каждая деталь представляет из себя плоский квадрат размера 1×1 . Квадраты можно соединять между собой, совмещая их стороны. Например, из 6 деталей можно собрать куб размера $1 \times 1 \times 1$, а из 22 деталей можно собрать прямоугольный параллелепипед размера $1 \times 2 \times 3$. Программа должна вычислять размеры сторон такого прямоугольного параллелепипеда, который можно собрать из деталей конструктора, и у которого будет максимально возможный объем.

Исходные данные

В первой строке содержится целое число t – количество тестов, $1 \leq t \leq 1000$. В следующих t строках содержатся целые числа n_1, n_2, \dots, n_t , в $(i+1)$ -й строке записано число n_i – количество деталей в конструкторе для i -го теста, $6 \leq n_i \leq 10^6$, $1 \leq i \leq t$.

Результат

Ваша программа должна вывести t строк. В i -й строке ($1 \leq i \leq t$) должны быть выведены три числа – размеры сторон такого прямоугольного параллелепипеда, который можно собрать из деталей конструктора с n_i деталями, и у которого будет максимально возможный объем. Если такой прямоугольный параллелепипед определен неоднозначно, выведите размеры сторон любого из них. Размеры сторон необходимо вывести в порядке возрастания в одну строку, разделяя их пробелами.

Пример

Исходные данные	Результат
3	1 1 1
6	1 2 3
22	1 2 3
23	