

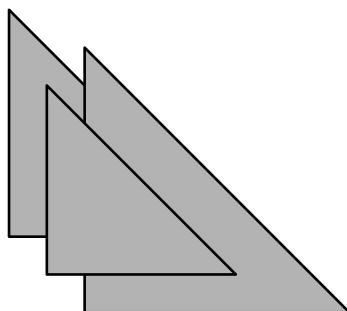
Задача А. Атака инопланетян

Имя входного файла: `attack.in`
Имя выходного файла: `attack.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На город Энск нападает флот инопланетян. Флот состоит из n космических кораблей, каждый из которых имеет форму равнобедренного прямоугольного треугольника.

Носом инопланетного корабля считается вершина, угол при которой прямой, а осью корабля называется высота, опущенная на гипотенузу.

Флот инопланетян прилетел с северо-востока, и застыл в таком положении, что все оси кораблей направлены строго на юго-запад.



Единственный способ нанести урон инопланетной армии — это пустить из некоторой точки поверхности Земли лазерный луч вертикально вверх. Пущенный так луч прожигает насквозь все вражеские корабли, через которые он проходит (даже те, которые он задевает по границе). Но этот выстрел повредит инопланетянам только в случае, если все n кораблей будут при этом поражены.

Военные власти города Энска решили нанести удар по вражеским войскам. Для этого решено поставить лазер в одну из точек, над которыми находятся все n вражеских кораблей.

Помогите военным определить площадь территории, на которой можно поставить лазер.

Формат входного файла

В первой строке входного файла содержится целое число n — количество инопланетных кораблей ($1 \leq n \leq 100$).

В каждой из следующих n строк описывается положение очередного корабля. Описание состоит из трех целых чисел x_i , y_i и s_i , где x_i и y_i — координаты носа, а s_i — размер корабля. Поскольку корабль имеет форму равнобедренного прямоугольного треугольника, размером корабля военные решили называть длину катета.

Размеры кораблей — положительные числа, не превышающие 1 000. Координаты носов кораблей не превышают по абсолютной величине 10^5 .

Формат выходного файла

В выходной файл выведите площадь территории, над которой находятся все инопланетные корабли. Выведите ответ с точностью до трех знаков после десятичной точки.

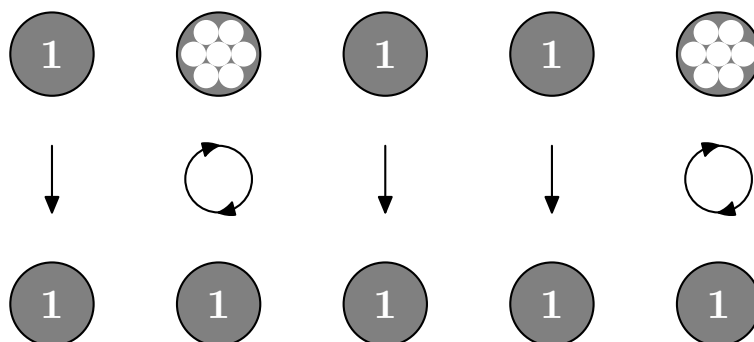
Пример

<code>attack.in</code>	<code>attack.out</code>
3 2 4 6 4 2 7 3 3 5	4.500

Задача В. Монетки

Имя входного файла: `coins.in`
Имя выходного файла: `coins.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На столе лежат n монеток. Некоторые из них лежат вверх решкой, а некоторые — гербом. Определите минимальное число монеток, которые нужно перевернуть, чтобы все монетки были повернуты вверх одной и той же стороной.



Формат входного файла

В первой строке входного файла содержится натуральное число n — количество монет ($1 \leq n \leq 100$).

В каждой из следующих n строк содержится одно целое число — 1 если монетка лежит вверх решкой или 0 если вверх гербом.

Формат выходного файла

В выходной файл выведите минимальное количество монет, которые нужно перевернуть.

Пример

coins.in	coins.out
5 1 0 1 1 0	2

Задача С. Белоснежка и n гномов

Имя входного файла: `dwarfs.in`
Имя выходного файла: `dwarfs.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

«Ну не гномы, а наказание какое-то!», — подумала Белоснежка, в очередной раз пытаясь уложить гномов спать. Одного уложишь — другой уже проснулся! И так всю ночь.

У Белоснежки n гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать i -го гнома нужно a_i минут, и после этого он будет спать ровно b_i минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома, $a_1 = 1$, $b_1 = 10$, $a_2 = 10$, $b_2 = 20$. Если Белоснежка сначала начнет укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснется первый. Если же она начнет со второго гнома, то затем она успеет уложить первого и получит целых 10 минут отдыха.

Формат входного файла

Первая строка входного файла содержит число n ($1 \leq n \leq 10^5$), вторая строка содержит числа a_1, a_2, \dots, a_n , третья — числа b_1, b_2, \dots, b_n ($1 \leq a_i, b_i \leq 10^9$).

Формат выходного файла

Выведите в выходной файл n чисел — порядок, в котором нужно укладывать гномов спать. Если Белоснежке отдохнуть не удастся, выведите число -1 .

Пример

dwarfs.in	dwarfs.out
2 1 10 10 20	2 1
2 10 10 10 10	-1

Задача D. Игра с графом

Имя входного файла: `graph.in`
 Имя выходного файла: `graph.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайта

Петя и Вася играют в очередную интересную игру. У них есть лист бумаги, на котором изображены n кружочков, помеченных числами от 1 до n . Участники по очереди рисуют стрелочки, соединяющие кружочки. При этом стрелочку из кружочка a в кружочек b разрешено проводить, если выполнены два условия:

1. еще нет стрелочки из a в b ;
2. нельзя дойти по стрелочкам из b в a .

Например, в позиции на рис. 1 можно поставить одну из трех стрелочек (рис. 2).

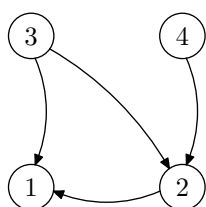


Рис. 1

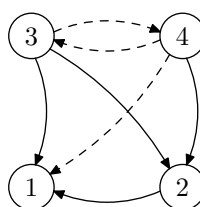


Рис. 2

Проигрывает тот, кто не может сделать ход.

Петя решил написать программу, играющую в эту игру. Для этого он хочет сначала посчитать, сколько различных позиций может получиться на доске.

Формат входного файла

Входной файл содержит одно число n ($1 \leq n \leq 100$).

Формат выходного файла

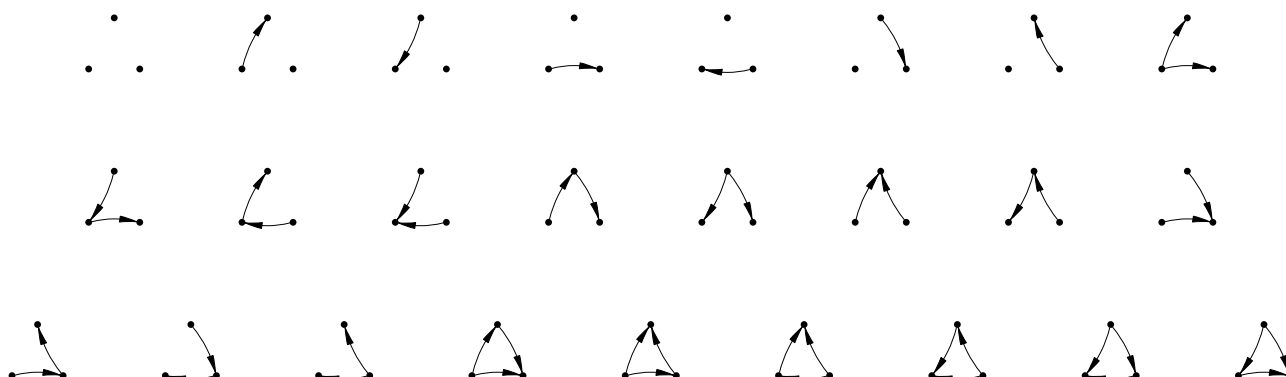
Выведите в выходной файл число возможных позиций без ведущих нулей.

Пример

<code>graph.in</code>	<code>graph.out</code>
3	25

Примечание

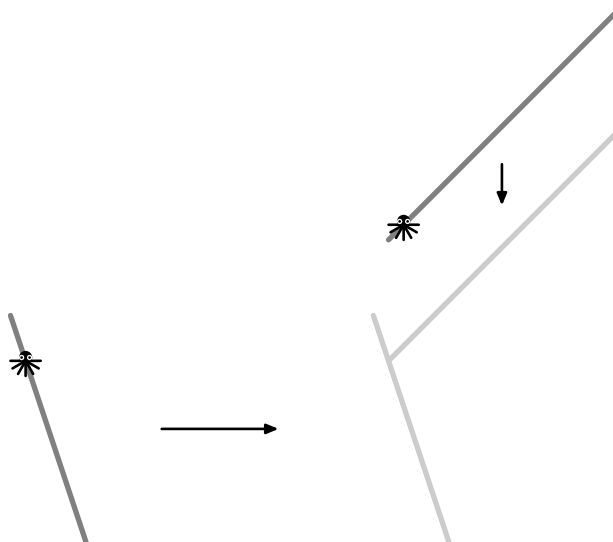
Приведем все 25 возможных позиций для примера из условия:



Задача Е. Про любовь...

Имя входного файла: `love.in`
Имя выходного файла: `love.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Паук и паучиха плывут по озеру на двух веточках. Плавать они не умеют, поэтому смогут встретиться только тогда, когда веточки соприкоснутся.



Считая, что веточки имеют форму отрезков, и что они плывут с постоянными скоростями, определите, сколько осталось ждать встречи несчастным членистоногим.

Формат входного файла

Входной файл содержит 12 чисел: $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, v_{1x}, v_{1y}, v_{2x}, v_{2y}$. Координаты вершин первого отрезка: (x_1, y_1) и (x_2, y_2) , координаты вершин второго отрезка: (x_3, y_3) и (x_4, y_4) , скорость первого отрезка (v_{1x}, v_{1y}) , скорость второго отрезка (v_{2x}, v_{2y}) . Все числа целые и не превосходят по модулю 10^4 . В начальный момент времени веточки не соприкасаются.

Гарантируется, что веточки имеют ненулевую длину.

Формат выходного файла

Выведите в выходной файл время до ближайшего момента, когда веточки соприкоснутся, с ошибкой не более 10^{-4} . Если веточки не соприкоснутся никогда, выведите число -1 .

Пример

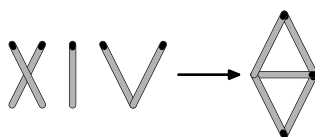
love.in	love.out
0 0 -1 3 4 4 7 7 3 0 0 -1	1.6
0 0 -1 3 4 4 7 7 1 0 0 -3	-1

Задача F. Спички детям не игрушка!

Имя входного файла: `matches.in`
Имя выходного файла: `matches.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вася любит решать головоломки со спичками. Чаще всего они формулируется следующим образом: дано изображение A , составленное из спичек; переложите в нем минимальное количество спичек так, чтобы получилось изображение B .

Например, из номера текущего командного чемпионата школьников Санкт-Петербурга по программированию, можно получить ромб с диагональю, переложив всего три спички.



Головоломки, которые решает Вася, всегда имеют решение. Это значит, что набор спичек, используемый в изображении A , совпадает с набором спичек, используемым в изображении B . Кроме того, в одном изображении никогда не встречаются две спички, у которых есть общий участок ненулевой длины (то есть спички могут пересекаться, но не могут накладываться друг на друга).

Вася устал решать головоломки вручную, и теперь он просит вас написать программу, которая будет решать головоломки за него. Программа будет получать описания изображений A и B и должна найти минимальное количество спичек, которые надо переложить в изображении A , чтобы полученная картинка получалась из B параллельным переносом.

Формат входного файла

В первой строке входного файла содержится целое число n — количество спичек в каждом из изображений ($1 \leq n \leq 1000$).

В следующих n строках записаны координаты концов спичек на изображении A . Спичка номер i описывается целыми числами $x_{1i}, y_{1i}, x_{2i}, y_{2i}$ — координатами ее концов. Следующие n строк содержат описание изображения B в таком же формате. Набор длин этих спичек совпадает с набором длин спичек с изображения A .

Все координаты по абсолютной величине не превосходят 10^4 . Все спички имеют ненулевую длину, то есть $x_{1i} \neq x_{2i}$ или $y_{1i} \neq y_{2i}$.

Формат выходного файла

Выведите в выходной файл минимальное количество спичек, которые следует переложить, чтобы изображение A совпало с изображением B , с точностью до параллельного переноса.

Пример

<code>matches.in</code>	<code>matches.out</code>
5 0 0 1 2 1 0 0 2 2 0 2 2 4 0 3 2 4 0 5 2 9 -1 10 1 10 1 9 3 8 1 10 1 8 1 9 -1 8 1 9 3	3

Задача G. Лабиринт

Имя входного файла: `maze.in`
Имя выходного файла: `maze.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Открыв глаза, Принц Персии обнаружил, что находится на верхнем уровне подземного лабиринта Джаффара. Лабиринт состоит из h уровней, расположенных строго друг под другом. Каждый уровень представляет собой прямоугольную площадку, разбитую на $m \times n$ участков. На некоторых участках стоят колонны, поддерживающие потолок, на такие участки Принц заходить не может.

Принц может перемещаться с одного участка на другой участок того же уровня, если у этих участков есть общая сторона, и ни один из этих участков не содержит колонну. Это перемещение занимает у Принца 5 секунд.

Полы в лабиринте Джаффара чрезвычайно тонкие, и Принцу не составляет труда сильным ударом ноги проломить пол под собой, если только на соответствующем участке нижнего уровня не находится колонна. Когда пол проламывается, Принц проваливается на один уровень вниз, при этом не перемещаясь в горизонтальной плоскости. Это действие также занимает у Принца 5 секунд. Конечно, если Принц уже находится на самом нижнем уровне, то пол под ним не проломится.

На одном из участков нижнего уровня Принца ждет Принцесса, отказавшаяся выйти замуж за злого Джаффара. Помогите Принцу найти Принцессу, потратив на это как можно меньше времени.

Формат входного файла

В первой строке входного файла содержатся натуральные числа h , m и n — высота и горизонтальные размеры лабиринта ($2 \leq h, m, n \leq 50$). Далее во входном файле приведены h блоков, описывающих уровни лабиринта в порядке от верхнего к нижнему.

Каждый блок содержит m строк, по n символов в каждой: «.» (точка) обозначает свободный участок, «o» (строчная латинская буква «o») обозначает участок с колонной, «1» обозначает свободный участок, в котором оказался Принц в начале своего путешествия, «2» обозначает свободный участок, на котором томится Принцесса.

Символы «1» и «2» встречаются во входном файле ровно по одному разу: символ «1» — в описании самого верхнего уровня, а символ «2» — в описании самого нижнего.

Соседние блоки разделены одной пустой строкой.

Формат выходного файла

В выходной файл выведите минимальное время в секундах, необходимое Принцу, чтобы найти Принцессу. Поскольку Добро всегда побеждает Зло, гарантируется, что Принц может это сделать.

Пример

maze.in	maze.out
3 3 3 1.. oo. ... ooo ..o .oo ooo o.. o.2	60

Задача Н. Число

Имя входного файла: `number.in`
Имя выходного файла: `number.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По заданным числам a , b , c и d , найдите наименьшее натуральное число n , большее ac , которое нельзя представить в виде произведения двух натуральных чисел u и v , таких что $a \leq u \leq b$ и $c \leq v \leq d$.

Формат входного файла

Входной файл содержит одну строку, состоящую из натуральных чисел a, b, c, d ($1 \leq a \leq b \leq 10^6$, $1 \leq c \leq d \leq 10^6$).

Формат выходного файла

Выведите в выходной файл искомое число n .

Пример

<code>number.in</code>	<code>number.out</code>
1 2 1 2	3
1 2 3 5	7

Задача I. Красивые номера

Имя входного файла: `phones.in`
Имя выходного файла: `phones.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вы, наверное, замечали, что многие компании используют для рекламы «красивые» номера телефонов, которые удобны для запоминания потенциальными клиентами. Но что делать, если номер вашей компании ничем не примечателен? Можно присмотреться к нему повнимательнее, а вдруг, если перегруппировать цифры номера некоторым образом, номер станет намного красивее? Например, если у вашей компании номер 872-73-33, то его можно сделать красивее, если перегруппировать цифры так: 8727-333.

Введем следующую оценку красоты разбиения номера. Будем разбивать номер дефисами на группы размером от 2 до 4 цифр. Теперь красотой разбиения назовем сумму баллов, которые приносит каждая группа. Эти баллы будем считать, пользуясь следующей таблицей.

Шаблон группы	Баллы
aa	2
aba	2
aab, abb	2
aaa	3
abac, baca	2
abab	3
aabb	3
abba	4
baaa, abaa, aaba, aaab	3
aaaa	5

В этой таблице символами «a», «b», «c» обозначены различные цифры. Например под шаблон «aab» подходят группы «223», «667», но не подходят «123» и «888».

Пользуясь предложенной оценкой, найдите наиболее красивое разбиение заданного номера.

Формат входного файла

Входной файл содержит одну строку из 7 цифр — заданный телефонный номер.

Формат выходного файла

Выведите в первой строке выходного файла наиболее красивое разбиение номера, а во второй — величину его красоты.

Если разбиений с максимальной величиной красоты несколько, выведите в выходной файл любое из этих разбиений.

Пример

<code>phones.in</code>	<code>phones.out</code>
8727333	8727-333 5
8827291	88-272-91 4

Задача J. Зоопарк

Имя входного файла: `zoo.in`
Имя выходного файла: `zoo.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В городском зоопарке содержатся животные n разных видов. Для участия в международной выставке «Три твари» зоопарк должен представить трех животных различных видов. Теперь служителей зоопарка интересует, сколькими способами можно выбрать трех животных для участия в выставке.

Например, если в зоопарке два медведя, тигр, лев и пингвин, то есть семь способов выбрать трех животных:

1. первый медведь, тигр и лев;
2. первый медведь, тигр и пингвин;
3. первый медведь, лев и пингвин;
4. второй медведь, тигр и лев;
5. второй медведь, тигр и пингвин;
6. второй медведь, лев и пингвин;
7. тигр, лев и пингвин.

Формат входного файла

В первой строке входного файла содержится натуральное число n — количество видов животных в городском зоопарке ($1 \leq n \leq 10^5$).

В каждой из следующих n строк содержится одно натуральное число — количество животных соответствующего вида. Общее число животных в зоопарке не превышает 10^5 .

Формат выходного файла

В выходной файл выведите количество способов выбрать трех животных для международной выставки.

Пример

<code>zoo.in</code>	<code>zoo.out</code>
4 2 1 1 1	7
3 30000 30000 30000	27000000000000