

Задача 1. Годовая оценка

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В школе учитель математики ввел новую систему выставления годовых оценок:

1. Ученику выставляется оценка 5, если все его оценки не ниже 4, а средний балл за все контрольные не ниже 4.7.
2. Оценку 4 можно получить, если среди оценок ученика нет двоек, а средний балл за все контрольные не ниже 3.6.
3. Итоговая оценка 3 выводится в том случае, если средний балл за все контрольные не ниже 2.3.
4. В остальных случаях ставится оценка 2.

Школьник Рома очень хочет получить хорошую годовую оценку (4 или 5). Он помнит количество баллов, полученные за все свои контрольные. Помогите Роме вычислить его годовую оценку.

Формат входного файла

В первой строке входного файла задано одно целое число n — количество оценок за контрольные, полученные в течение года ($1 \leq n \leq 10^5$).

Во второй строке через пробел записано n чисел, где i -е число — количество баллов за i -ю контрольную. Каждая контрольная оценивается по 100-балльной системе.

В третьей строке задано три целых числа, определяющих, сколько нужно набрать баллов за контрольную, чтобы получить по ней оценки 5, 4 или 3 соответственно.

Формат выходного файла

В первую строку выходного файла нужно вывести одно целое число — итоговую оценку, которую получит Рома.

Во вторую строку необходимо записать вещественное число — полученный средний балл. Ответ будет считаться верным, если он будет отличаться от значения правильного среднего балла не более чем на 10^{-6} .

Пример

| <code>input.txt</code> | <code>output.txt</code> |
|-----------------------------------|-------------------------|
| 5 65 100 70 100 80 85 50 30 | 4 4.400000 |

Задача 2. Жадные гномики

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

N гномов долго трудились в своих пещерах, ища драгоценные камни. Добыв ровно N камней, они пошли в дом волшебника, чтобы размножить там свое богатство. Гномы сложили все свои камни в кучу, чтобы волшебник прочитал свое заклинание. После каждого заклинания количество камней увеличивается вдвое.

Но есть одна беда: у волшебника в доме живет злобный карлик, который сразу после каждого заклинания забирает из кучи ровно один камень. То есть, если до заклинания в куче было x камней, то после заклинания их станет $2x - 1$.

Гномы очень жадные и ревнивые создания, поэтому они хотят, чтобы все камни разделились между ними поровну. Если этого не произойдет, то они заставят волшебника снова прочесть свое заклинание. И снова, и снова, пока количество камней не станет кратным N . А карлик при этом не дремлет и продолжает красть камни.

Вопрос: закончит ли волшебник свое дело, и, если да, то сколько заклинаний ему придется прочесть?

Формат входного файла

Во входном файле записано одно целое число N — количество гномиков ($1 \leq N \leq 2 \cdot 10^9$).

Формат выходного файла

В выходной файл нужно вывести одно целое число — необходимое количество заклинаний, после которых число драгоценных камней станет кратным N . Если такого числа нет, то вывести -1 .

Примеры

| <code>input.txt</code> | <code>output.txt</code> |
|------------------------|-------------------------|
| 1 | 1 |
| 2 | -1 |
| 7 | 3 |

Задача 3. Реми и кулинарный поединок

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда (2 секунды для Java)
Ограничение по памяти: 256 мегабайт

Известного крысёнка Реми, первоклассного шеф-повара, известного по мультфильму «Рататуй», пригласили поучаствовать в одном престижном кулинарном телешоу.

Каждый участник шоу за ограниченное время должен приготовить N блюд и представить их судьям Международной Дегустационной Коллегии (МДК), состоящей из внушительного числа людей, разбирающихся в искусстве кулинарии.

Приготовление блюд не вызвало у Реми никаких затруднений, и он уже выставил все N тарелок на длинный стол, вдоль которого будут проходить выбранные Реми члены МДК, каждый из которых должен совершить процедуру обобщённого кушанья. Член жюри проходит вдоль стола слева направо и, как только видит еще никем не опробованное блюдо, дегустирует его и следующие за ним блюда. Реми готовит так вкусно, что ни один не сможет просто пройти мимо блюда, не попробовав его.

Процедура обычного кушанья часто состоит из поедания «первого», «второго», и, возможно, компота. Процедура же **обобщенного кушанья** состоит из поедания «первого», «второго», ..., «пятого», ... , «десятого» и т.д. И вообще эта процедура может состоять из любого непустого набора блюд, чьи типы образуют **монотонную** (неубывающую или невозрастающую) последовательность. Реми считает, что ставить два блюда одного типа подряд — дурной тон и безвкусица, поэтому гарантируется, что типы соседних блюд на столе различны.

Но какой же обед без компота? Поэтому после каждой последовательности блюд, Реми хочет поставить компотик, который бы радовал жюри и символизировал, что обед для этого члена жюри окончен. Реми хочет узнать, сколькими способами он может расставить бокалы с компотом между блюдами так, чтобы удовлетворить всех членов жюри, которые будут проходить мимо его стола. Чтобы избежать претензий от конкурентов о подкупе членов МДК, Реми решил, что не стоит ставить больше одного бокала с компотом после каждой последовательности, образующей обобщённое кушанье.

Формат входного файла

В первой строке входного файла задано целое число N — количество блюд, приготовленных Реми ($1 \leq N \leq 5 \cdot 10^5$).

В следующей строке через пробел записаны N целых чисел t_i — типы блюд в том порядке, в котором они выставлены на столе, начиная с левого края ($1 \leq t_i \leq 10^9$).

Формат выходного файла

В выходной файл необходимо вывести одно целое число — количество способов расставить бокалы с компотом так, чтоб удовлетворить всех проходящих мимо членов МДК. Так как число может быть очень большим, то нужно вывести остаток от деления его на $10^9 + 7$.

Примеры

| <code>input.txt</code> | <code>output.txt</code> |
|------------------------|-------------------------|
| 3 2 1 3 | 3 |
| 4 1 3 4 2 | 6 |

Задача 4. Взволнованный водитель

| | |
|-------------------------|-------------------------|
| Имя входного файла: | <code>input.txt</code> |
| Имя выходного файла: | <code>output.txt</code> |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Крот Афанасий на этой неделе, наконец, сдал экзамен на водительское удостоверение. Теперь он может сесть за руль своего роскошного автомобиля и поехать куда пожелает. Перед ним открыты просторы всего мира, и ему уже не терпится испытать свой автомобиль и водительские навыки в действии.

На первый раз Афанасий решил ограничиться путешествием по прямой линии с постоянной скоростью. Как и любой начинающий водитель, Афанасий чувствует себя немного неуверенно. В частности, он беспокоится, не превысит ли он ненароком ограничение скорости. Поэтому он просит вас, как самого близкого друга, помочь ему.

Для упрощения будем считать город бесконечной двумерной плоскостью. Афанасию известно, что превышение скорости в городе контролируется M камерами. Все камеры пронумерованы числами от 1 до M . Камера с номером i расположена в точке (x_i, y_i) и смотрит в направлении точки $(x_i + dx_i, y_i + dy_i)$. Она может поворачиваться от этого направления вправо и влево на 90° .

В определенные моменты времени Афанасий хочет знать, какое количество камер контролирует его движение. Помогите Афанасию, отвечая на его запросы.

Формат входного файла

В первой строке входного файла задано целое число M — количество камер в городе ($1 \leq M \leq 10^5$).

В следующих M строках дано описание камер. Каждая камера описывается четырьмя целыми числами x_i, y_i, dx_i, dy_i — координатами камеры и направляющим вектором обзорной полуплоскости ($-10^9 \leq x_i, y_i, dx_i, dy_i \leq 10^9$).

В следующей строке записано целое число N — количество точек, находясь в которых, Афанасий волнуется и просит вас узнать количество следящих за ним камер ($2 \leq N \leq 10^5$).

Далее в N строках даны координаты этих точек, по два числа в строке. Каждая координата по модулю не превосходит 10^9 .

Гарантируется, что все точки попарно различны и лежат на одной прямой, по которой и путешествовал Афанасий. Причем точки заданы именно в том порядке, в котором их посещал водитель.

Формат выходного файла

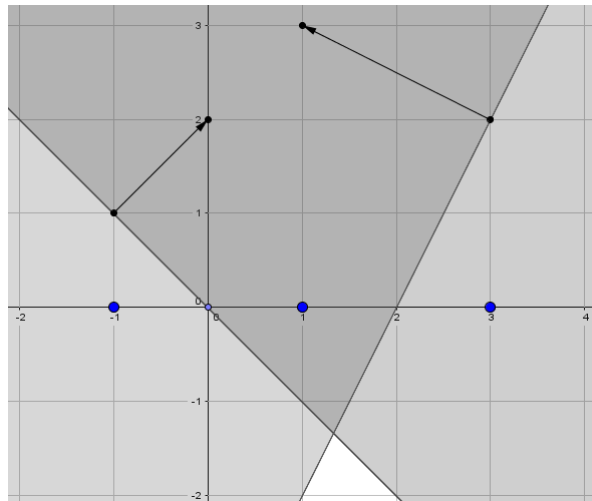
В выходной файл необходимо вывести N строк. В каждой строке должно быть записано одно целое число — количество камер, наблюдавших за Афанасием, когда он находился в соответствующей точке. Эти числа нужно выводить в том же порядке, в каком заданы точки во входном файле.

Примеры

| input.txt | output.txt |
|--|-------------|
| 2 -1 1 1 1 3 2 -2 1 3 -1 0 1 0 3 0 | 1 2 1 |
| 1 5 3 -8 1 2 -2 0 7 1 | 1 0 |

Пояснение к примеру

На рисунке показано расположение камер в первом тесте. Каждая из камер обозревает свою выделенную область, включая «границу» (прямую, перпендикулярную изображённому вектору).



Задача 5. Петя и IT

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Петя — способный, но очень неусидчивый школьник. Как обычно, закончив своё задание на уроке компьютерной графики намного быстрее своих товарищей, он решил скоротать время за своей любимой игрой “Insane Traveller” (IT). Главная цель игры — как можно скорее завершить путешествие из начального города в конечный. Петя застрял на одном непростом уровне, и просит вас помочь ему завершить миссию.

В игре имеется N городов и M дорог между городами, по которым можно перемещаться в любом направлении. Путешествие по любой из дорог занимает ровно одни сутки. Главный герой обладает одной очень полезной сверхспособностью: находясь в городе с порядковым номером k игрок может запустить процесс «мирового переворота». В результате этого процесса уничтожаются все существующие дороги и достраиваются дороги, которых раньше не существовало. Эта операция занимает t_k суток, в течение которых герой должен оставаться в городе, из которого был запущен процесс. Игрок имеет право запускать процесс «мирового переворота» **не более одного раза** за всё своё путешествие.

Напомним, что требуется как можно скорее оказаться в пункте назначения — городе с порядковым номером N , если вы стартуете из города с порядковым номером 1.

Формат входного файла

В первой строке входного файла через пробел заданы два целых числа N и M — количество городов и дорог, соответственно ($1 \leq N, M \leq 10^6$).

В следующей строке через пробел записаны N целых чисел t_i — время, требующееся на проведение «мирового переворота» из города с номером i ($1 \leq i \leq N, 1 \leq t_i \leq 10^6$).

Далее в M строках содержится описание дорог: в i -й строке через пробел записано два целых числа u_i, v_i — порядковые номера городов, соединенных i -й дорогой ($1 \leq u_i, v_i \leq N$). Гарантируется, что между любой парой городов существует не более одной прямой дороги, и что не существует дорог, соединяющих город с самим собой.

Формат выходного файла

В выходной файл необходимо вывести одно целое число — наименьшее количество суток, необходимых для совершения путешествия из начального города в конечный. Если путешествие совершить невозможно, нужно вывести число -1 .

Пример

| input.txt | output.txt |
|--|------------|
| 5 3 100 100 100 50 100 1 2 2 3 3 4 | 54 |