

Задача А. Шоколад

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Малыш обиделся на Карлсона. Тот без спроса съел в доме всё сладкое и даже к соседям залезть успел. И вновь — только родители принесли домой большую шоколадку размера $m \times n$, а Карлсон тут как тут! Малыш твёрдо решил, что на этот раз ни кусочка шоколада этому обжоре не достанется. Известно, что кроме еды у Карлсона есть ещё одна слабость: он любит играть в азартные игры и иногда ради возможности сыграть даже готов пожертвовать своим желудком. Поэтому Малыш предложил Карлсону следующую игру: за один ход игрок должен взять один из кусков шоколада и разломить его на два куска вдоль линий, разделяющих дольки. Игроки ходят по очереди; проигрывает тот, кто не может сделать ход, так как остались только куски шоколада размера 1×1 . Победитель забирает весь шоколад себе.

Однако Карлсона так просто не проведёшь! Он быстро смекнул, должен ли он ходить первым или уступить право первого хода Малышу, чтобы гарантированно выиграть. А Вы бы смогли определить это на его месте?

Формат входного файла

В единственной строке через пробел записаны целые числа m и n ($1 \leq m, n \leq 50$) — длина и ширина шоколадки в дольках.

Формат выходного файла

Если для того, чтобы выиграть, Карлсону нужно ходить первым, выведите в единственной строке «`[:=[first]`», иначе выведите «`[second]=:`».

Примеры

<code>input.txt</code>	<code>output.txt</code>
2 4	<code>[:=[first]</code>
1 3	<code>[second]=:</code>

Задача В. Кольцо холода

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Лич Сандро объявил войну королю Ада. И теперь полчища демонов ведут ожесточённую битву против армии нежити Сандро. Сандро в совершенстве владеет магией огня, но созданиям Ада огонь не причиняет никакого урона. Поэтому Сандро решил использовать против них заклинание «Кольцо холода». Это заклинание создаёт на поле боя непробиваемую ледяную стену в форме окружности. Демоны, которых задела ледяная стена, мгновенно погибают, а те демоны, которые оказались окружёнными стеной, хоть и остаются живы, более не могут принимать участия в битве.

Сандро может мгновенно телепортироваться в любую точку поля боя и применить в ней «Кольцо холода». В этом случае возникнет ледяная стена с центром в той точке, где будет находиться Сандро. Учтите, что магии Сандро хватит на создание стены радиусом не более 10 000 метров. Теперь он хочет так выбрать радиус заклинания и точку, куда он должен телепортироваться, чтобы хотя бы один демон был убит, а все демоны, оставшиеся в живых, были заключены внутрь стены.

Формат входного файла

Первая строка содержит целое число n — количество демонов ($1 \leq n \leq 100$). На поле боя демонов можно считать точками. В следующих n строках перечислены пары целых чисел (x, y) — координаты демонов относительно той точки, где изначально находится Сандро. Координаты измеряются в метрах и не превышают по модулю 1 000. В каждой точке поля находится не более одного демона; также демон не может находиться в той точке, где стоит Сандро.

Формат выходного файла

Выведите три действительных числа с точностью не менее 10^{-9} — координаты точки, куда должен телепортироваться Сандро, и радиус «Кольца холода», которое он должен создать. Сандро не может телепортироваться в точку, где в данный момент находится некоторый демон. Гарантируется, что задача всегда имеет решение.

Примеры

<code>input.txt</code>	<code>output.txt</code>
7 1 1 1 5 3 6 5 3 8 0 9 5 5 9	5 4 5
2 0 2 2 0	1 1 1.41421356237309

Задача С. Дежурства

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Директор школы решил внедрить прогрессивную систему дежурств для поддержания порядка во всех кабинетах школы. Каждому ученику заранее назначается кабинет, за который он отвечает. Каждый день по школе должны дежурить два человека, отвечающие за разные кабинеты. Ученики в день своего дежурства должны полить цветы в тех кабинетах, за которые они отвечают, а также проследить, чтобы во всех кабинетах информатики были невысохшие маркеры для доски. Директор поручил вам составить план дежурств на первые m учебных дней. Все ученики за это время должны хотя бы по разу побывать дежурными. План должен определить для каждого ученика, за какой кабинет он отвечает и в какие дни наступает его очередь быть дежурным. Разумеется, за каждый кабинет школы должен отвечать кто-то из учеников. Директор дополнительно требует, чтобы в составленном вами плане никакая пара учеников не дежурила дважды. Чтобы облегчить вам задачу, директор разрешил распределять дежурства неравномерно — провинившимся ученикам можно поручить большее количество дежурств, чем прилежным.

Формат входного файла

В первой строке через пробел записаны три целых числа: n — количество учеников в школе, k — количество кабинетов и m — количество дней в требуемом плане дежурств. Числа удовлетворяют ограничениям $2 \leq k \leq n \leq 100$. Можете считать, что при заданных n , k и m всегда существует хотя бы один план дежурств.

Формат выходного файла

Занумеруем учеников целыми числами от 1 до n , а кабинеты — целыми числами от 1 до k . В первых n строках выведите распределение кабинетов по дежурным: в i -й строке должен быть записан номер кабинета, за который будет отвечать ученик с номером i . Следующие m строк должны содержать пары номеров дежурных в каждый из m учебных дней. Помните, что пары не должны повторяться и в одну пару нельзя ставить учеников, отвечающих за один и тот же кабинет.

Примеры

input.txt	output.txt
5 3 4	1 2 1 2 3 1 2 3 4 3 5 4 5
4 2 4	1 2 1 2 1 2 3 4 1 4 2 3

Задача D. Одномерный лабиринт

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

В одномерном государстве жили одномерные люди. И всё остальное в этом государстве было тоже одномерное. Всё в их одномерном мире было просто и понятно: всего одна ось координат и всего два направления движения — вперёд и назад. Но даже в одномерном мире есть проблемы. Например, как найти выход из лабиринта? Это нам одномерный лабиринт кажется по меньшей мере странным, а для одномерных людей поиск выхода из лабиринта — очень сложная и насущная задача. Решают её одномерные люди следующим образом.

Одномерный человек выбирает направление движения: вперёд (в сторону увеличения координаты) или назад (в сторону её уменьшения) и идёт в выбранном направлении. Если он находит точку выхода, то сразу покидает лабиринт. Если же он наталкивается на препятствие, то разворачивается и идёт в обратном направлении.

Почувствуйте, как нелегко жить в одномерном мире: попробуйте реализовать функцию, определяющую расстояние, которое пройдёт одномерный человек, прежде чем найдёт выход, в зависимости от выбора начального направления движения.

Формат входного файла

В первой строке через пробел записаны два целых числа n и x — количество препятствий и координата точки выхода, соответственно ($0 \leq n \leq 100$). Одномерный человек располагается в начале координат. Во второй строке через пробел записаны n различных целых чисел — координаты препятствий. Все координаты, включая x , не равны нулю и не превосходят по модулю 1000. Гарантируется, что ни одно препятствие не расположено в точке выхода. Гарантируется, что вне зависимости от выбора начального направления движения одномерный житель рано или поздно найдёт выход или натолкнётся на препятствие.

Формат выходного файла

Выведите через пробел два числа — расстояние, которое пройдёт одномерный человек до выхода из лабиринта, если сначала пойдёт вперёд и если сначала он пойдёт назад, соответственно. Если из-за препятствий дойти до выхода невозможно, выведите «Impossible».

Примеры

<code>input.txt</code>	<code>output.txt</code>
3 -2 -10 -4 2	6 2
3 -2 10 -1 2	Impossible

Задача E. Атака Тёмной крепости

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Узнав, что лич Сандро ушёл воевать с королём демонов, эрафийские военачальники решили воспользоваться его отсутствием и захватить Тёмную крепость. Из Стедвика вышла армия крестоносцев во главе с Катериной Айронфист. В тот же день из лесов АвЛи им на помощь вышла армия эльфов-снайперов под предводительством легендарного стрелка Джелу. Военачальники прекрасно понимают, что пехота уязвима для личей, а стрелы эльфов крайне неэффективны против скелетов, поэтому двум армиям нужно объединиться и напасть на крепость одновременно. И поскольку Сандро может вернуться в любой момент, Тёмная крепость должна быть атакована как можно раньше. Вас же, как придворного картографа, попросили посчитать, какое минимальное количество дней уйдёт на реализацию этого плана.

Эрафия поделена на квадратные районы. Некоторые из них проходимы, некоторые — нет. За один день каждая армия может перейти в любой из 8 соседних (имеющих общую вершину) районов, если он проходим. В некоторых проходимых районах Эрафии установлен телепорт одного из 26 типов. Тип телепорта обозначается заглавной латинской буквой. Если армия располагается в районе с телепортом, то она может мгновенно переместиться в любой район с телепортом того же типа. В тот момент, когда обе армии оказываются в одном районе, они объединяются и далее перемещаются как единое целое. До объединения ни одна из армий не может нападать на крепость Сандро, то есть делать ход в район, в котором расположена крепость.

Формат входного файла

В первой строке через пробел записаны числа n и m — размеры карты Эрафии ($1 \leq n, m \leq 100$). Далее задана сама карта — n строк по m символов в каждой. Символы имеют следующий смысл:

- «#» — непроходимый район.
- «.» — проходимый район.
- «A»...«Z» — телепорт соответствующего типа.
- «\$» — армия Катерины.
- «!» — армия Джелу.
- «*» — крепость Сандро.

Гарантируется, что символы «*», «\$», «!» встречаются на карте ровно один раз.

Формат выходного файла

Выведите единственное число — минимальное количество дней, через которое объединённая армия сможет напасть на крепость Сандро. Если нападение совершить невозможно, выведите «Impossible».

Примеры

input.txt	output.txt
5 8 ...AA.# #####!*## ##### ..B\$...B	11
3 5 ##*.. !#... ##..\$	Impossible

Задача F. Куча орехов

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Десять больших грецких орехов — это, конечно, целая куча! А два ореха — совсем мало, ни о какой куче здесь не может идти и речи. А как насчёт шести орехов? Много это или мало? Попугай между тем уже давно ответил для себя на этот вопрос: «Мало — это когда всё съел и ещё хочется, а куча — это когда больше уже не хочется.» Попугай любит покушать, но не любит переедать. Поэтому он решил узнать минимальное количество орехов, которого ему хватит для того, чтобы насытиться.

С этой целью он провёл серию экспериментов. Каждый эксперимент заключался в следующем: изрядно проголодавшись, Попугай в один присест съедает некоторое количество орехов и записывал, наелся ли он этим количеством. Конечно, если Попугай наелся некоторым количеством орехов, то он должен наестся и любым бóльшим количеством. И наоборот, если он остался голодным, съев несколько орехов, то он должен остаться голодным, съев и любое меньшее количество орехов. Помогите Попугаю обработать полученные результаты.

Формат входного файла

В первой строке записано целое число n — количество экспериментов, проведённых Попугаем ($0 \leq n \leq 100$). Каждая из следующих n строк содержит описание очередного эксперимента — количество орехов, съеденных Попугаем (целое число от 3 до 9), и через пробел слово «hungry» или «satisfied», указывающее, остался ли Попугай голодным или наелся, соответственно. Известно, что Попугай заведомо наестся десятью орехами и заведомо останется голодным, съев лишь два.

Формат выходного файла

Выведите минимальное количество орехов, которое должен съесть Попугай, чтобы гарантированно не остаться голодным. Если же записи Попугая противоречивы, выведите «Inconsistent».

Примеры

input.txt	output.txt
4 4 hungry 7 satisfied 5 hungry 8 satisfied	7
2 6 hungry 5 satisfied	Inconsistent

Задача G. Лыжная гонка

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

На этот раз к вам за помощью обратились организаторы Зимних Олимпийских игр 2014 года, которые, как известно, пройдут в городе Екатеринозаводске. И хотя до Олимпийских игр ещё целых пять с половиной лет, первый спортивный объект уже сдан в эксплуатацию. Им стала трасса для проведения соревнований по лыжным гонкам.

Несмотря на то, что на трассе установлено только новое и надёжное оборудование, организаторы хотят разработать план действий в случае его сбоя. Что, например, делать, если из-за поломки секундомера на финише можно будет определить лишь порядок, в котором финишировали спортсмены? Дело осложняется правилами, по которым проводятся лыжные гонки с отдельным стартом. Участники стартуют по очереди, с интервалом в 30 секунд, поэтому участник, финишировавший первым, не обязательно будет первым в итоговой таблице результатов. Например, если спортсмен, стартовавший вторым, придёт к финишу на 25 секунд позже стартовавшего первым, то это значит, что он прошёл дистанцию на 5 секунд быстрее, а значит, должен располагаться в итоговой таблице выше.

Вам поручили написать программу, которая по порядку, в котором финишировали спортсмены, вычислит для каждого из них самое высокое и самое низкое место, на котором он может оказаться в итоговой таблице.

Формат входного файла

В первой строке записано целое число n — количество участников гонки ($1 \leq n \leq 2000$). Участники нумеруются целыми числами от 1 до n в том порядке, в котором они стартовали. Во второй строке записана перестановка чисел от 1 до n — порядок, в котором лыжники пришли на финиш. Числа в строке разделены пробелом.

Формат выходного файла

Выведите n строк, в i -й строке должны быть записаны через пробел два числа — самое высокое и самое низкое место, которое мог занять i -й участник.

Пример

input.txt	output.txt
6	3 6
3 5 1 4 2 6	4 6
	1 4
	2 5
	1 3
	1 6

Задача Н. Годзилла возвращается!

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

В 2567 году учёные из НИИ Цитологии и Генетики решили создать Багзиллу — огромного ящера, обладающего чудовищной силой. Как известно, рост и силу ящера целиком и полностью определяет его ДНК — специальная строка, состоящая из символов «А», «С», «G», «Т». В первые 2 дня учёные получили большое количество образцов ДНК двух ящеров-мутантов, которые впоследствии собирались использовать для получения ДНК Багзиллы. Каждый день они учились склеивать два ранее полученных образца ДНК, тем самым получая новый образец в количестве, достаточном для проведения дальнейших экспериментов. Наконец, через n дней они достигли желаемого результата — была получена ДНК ящера необычайного роста и силы. Информация об этом просочилась в прессу, и вот тут-то в городе началась паника. Уж если легендарный Годзилла в своё время камня на камне не оставил от города, то чего же ждать от его копии, ещё более чудовищной? Чтобы выяснить это, необходимо как можно быстрее посчитать, сколько раз ДНК Годзиллы встречается как подстрока в ДНК Багзиллы.

Формат входного файла

В первой строке записана непустая строка, состоящая только из символов «А», «С», «G», «Т», — образец ДНК Годзиллы, хранящийся в музее НИИ Цитологии и Генетики. Длина этой строки не превосходит 100 символов. В двух следующих строках записаны образцы ДНК ящеров, с которых начался процесс клонирования. Эти образцы представляют собой непустые строки длиной не более 10, также состоящие из символов «А», «С», «G», «Т». В четвёртой строке записано целое число n ($1 \leq n \leq 100$) — количество полученных учёными образцов ДНК (без учёта первых двух). Далее следуют n строк, k -я из них содержит пару целых чисел в пределах от 1 до $k+1$ — номера образцов ДНК, которые учёные научились склеивать в $(k+2)$ -й день экспериментов. Новая ДНК получается приписыванием второй ДНК в конец первой.

Формат выходного файла

Найдите число вхождений ДНК Годзиллы в ДНК Багзиллы и выведите его по модулю $10^9 + 7$.

Примеры

<code>input.txt</code>	<code>output.txt</code>
AGA A G 4 1 2 2 1 3 1 5 4	2
A СТ AG 2 1 1 3 3	0

В первом примере учёными были получены последовательно следующие образцы ДНК: «AG», «GA», «AGA», «AGAGA». ДНК «AGAGA» дважды содержит ДНК «AGA» как подстроку.

Во втором примере были получены последовательно следующие образцы ДНК: «СТСТ», «СТСТСТСТ». ДНК «СТСТСТСТ» не содержит ДНК «А».

Задача I. Раздел острова

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Необитаемый остров, на котором живёт Робинзон Крузо со своим верным слугой Пятницей, имеет форму невырожденного треугольника с вершинами в точках (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Однажды Робинзон и Пятница поссорились и решили поделить свой остров поровну, выбрав две точки на побережье острова и соединив их отрезком прямой. После раздела обе части острова должны были иметь одинаковую площадь и длину побережья. Робинзон так и не смог выбрать две нужные для этого точки. А сможете ли Вы?

Формат входного файла

В единственной строке записаны через пробел целые числа $x_1, y_1, x_2, y_2, x_3, y_3$, по модулю не превосходящие 2000.

Формат выходного файла

Если существует отрезок ST , делящий остров на части с равной площадью и длиной побережья, в первой строке выведите «YES», во второй строке — координаты точки S , в третьей — координаты точки T . Точки S и T должны лежать на побережье острова. Координаты следует выводить с точностью не менее 10^{-9} .

Если поделить остров требуемым образом невозможно, выведите в единственной строке «NO».

Примеры

input.txt	output.txt
0 0 10 0 0 10	YES 0 0 5 5
0 3 4 0 3 4	YES 1.741248277008306 3.580416092336102 3.445803840397070 0.415647119702198

Задача J. Яхты

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`

Всем известно, что лучшие в мире яхты строит знаменитая Екатеринозаводская верфь. Они настолько популярны, что как только богатый промышленник становится миллиардером, он в конце того же месяца приходит на верфь за яхтой. Ещё бы! На Екатеринозаводской верфи яхты изготавливаются вручную, в оформлении интерьера используются ценные породы красно-чёрного дерева. . . К сожалению, на верфи совсем немного рабочих, поэтому за месяц она может построить не более d новых яхт. В результате, верфь не всегда успевает справиться с наплывом покупателей. А миллиардеры — люди гордые: если они придут на верфь, а готовой яхты там не будет, то они обидятся и раздумают совершать покупку. Конечно, верфь может производить яхты впрок, но хранение готовых яхт недёшево — один месяц хранения обходится верфи в 1 золотой слиток.

Руководство решило узнать, какое максимальное количество яхт сможет продать верфь в ближайшие n месяцев и какое минимальное количество золота для этого придётся потратить на хранение готовых яхт. С этой целью руководство обратилось к студентам экономического факультета Екатеринозаводского университета, и те, исследовав динамику роста цен на полезные ископаемые, посчитали, сколько людей станет миллиардерами в каждый из n ближайших месяцев. Ваша же цель — ответить на вопросы руководства, используя эту информацию.

Формат входного файла

В первой строке через пробел записаны целые числа n и d ($1 \leq n \leq 20\,000$; $1 \leq d \leq 100\,000$). Во второй строке через пробел записаны целые числа a_1, a_2, \dots, a_n . Число a_i обозначает количество людей, которые станут миллиардерами в i -й месяц ($0 \leq a_i \leq 100\,000$).

Формат выходного файла

Выведите через пробел два числа — максимальное количество яхт, которое сможет продать верфь, и минимальное количество золотых слитков, которое для этого придётся потратить на хранение готовых яхт.

Пример

<code>input.txt</code>	<code>output.txt</code>
3 5 6 1 7	13 2