

Для всех задач:

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по памяти: 256 М
Максимальная оценка: 100 баллов

Задача 1. Опять двойка!

Ограничение по времени: 1 секунда на тест

Во входном файле задано N целых чисел, каждое из которых находится в диапазоне $[1..2^{31}-1]$. Подсчитайте количество чисел, встречающихся в этом файле, которые можно представить в виде 2^k , где k — целое число.

Входные данные

В первой строке входного файла записано одно целое число N ($0 < N \leq 10^6$).

Во второй строке — N целых чисел, разделенных пробелами.

Выходные данные

В выходной файл нужно записать одно целое число — количество чисел, удовлетворяющих условию задачи.

Пример

input.txt	output.txt
10 2 4 16 100000 5 3 8 2048 16 16	7

Задача 2. Система охраны

Ограничение по времени: 1 секунда на тест

Некоторое царство-государство окружено крепостной стеной с башнями, в которых расположены дозорные. Для оперативного оповещения каждого дозорного между башнями решили установить прямую телефонную. Ограниченная стеной территория царства-государства представляет собой многоугольник произвольного вида, башни располагаются в его вершинах.

Можно ли построить такую прямую связь, соединяющую каждую башню с каждой, чтобы кабель не проходил вне территории царства-государства.

Входные данные

В первой строке входного файла записано одно целое число — количество тестов, не более 10. Описание каждого теста состоит из двух строк, в первой строке указано одно целое число — количество башен, не превышающее 1000. Во второй строке через пробел перечислены целочисленные координаты башен по порядку их следования друг за другом, начиная с произвольной. Для каждой башни сначала указывается координата x , а затем y . Координаты по модулю не превосходят 10^5 . Считаем, что первая из перечисленных башен всегда соединена крепостной стеной с последней.

Выходные данные

В выходной файл для каждого теста на отдельной строке необходимо вывести слово **yes**, если возможно построить такую связь, или **no** в случае невозможности таковой.

Примеры

input.txt	output.txt
1 5 1 1 2 3 4 4 5 2 3 0	yes

1	no
5	
1 1 2 4 3 2 5 5 6 1	

Задача 3. Экономисты

Ограничение по времени: 1 секунда на тест

Требуется смонтировать локальную сеть. Вы, оценив обстановку и проведя замеры, получили данные о необходимом количестве отрезков кабеля, а также их длинах.

Кабель продается в катушках по 305м. Не всегда удастся разрезать кабель так, чтобы использовать катушку полностью.

Необходимо разрезать провод таким образом, чтобы сумма длин обрезков кабеля была минимальной, а также найти количество катушек провода, которое необходимо для монтажа сети.

Входные данные

В первой строке входного файла записано одно целое число n — необходимое количество отрезков кабеля ($0 < n \leq 10000$). Во второй строке дана последовательность из n целых чисел, разделенных пробелом — длины отрезков кабеля.

Выходные данные

Первая строка выходного файла должна содержать одно целое число — необходимое количество катушек. Во вторую строку записать найденную минимальную сумму длин обрезков кабеля.

Пример

<i>input.txt</i>	<i>output.txt</i>
3 100 102 101	1 2

Задача 4. Дорожно-транспортное происшествие

Ограничение по времени: 1 секунда на тест

Произошло ДТП. Машина скрылась с места происшествия. Свидетелем оказался математик. Он не смог запомнить номер машины, но сообщил следующую информацию о нем:

- четырёхзначный номер состоит из двух цифр A и B (например, $AABV$);
- сумма этих двух цифр равна S ($S = A + B$);
- номер делится на три различных простых числа.

Нужно восстановить номер автомобиля.

Входные данные

В первой строке входного файла записано одно целое число S ($0 \leq S \leq 18$).

Во второй строке записаны через пробел три целых простых числа p_1 , p_2 и p_3 , на которые делится номер.

Выходные данные

В выходной файл нужно записать одно целое число N — четырёхзначный номер автомобиля.

Пример

<i>input.txt</i>	<i>output.txt</i>
15 7 11 19	8778

Задача 5. Путешествие Емели

Ограничение по времени: 1 секунда на тест

Емеля, женившись на Несмеяне, получил полцарства в придачу. Захотелось ему осмотреть свои владения. Открыл он карту. Сколько на ней деревень, поселков и городов обозначено! Хочется ему как можно большее количество этих населенных пунктов посетить. Только вот одна проблема: хоть и быстроходная печь у Емели и вездеходная, но имеет один маленький конструктивный недостаток — не умеет печка поворачивать. “Ничего, — подумал Емеля, — доеду я до какой-нибудь окраинной деревеньки, там разверну печку, как надо, а затем напрямик через все государство прокачусь!”

Помогите Емеле посчитать максимальное количество населенных пунктов, через которые он может проехать, не сворачивая.

Входные данные

Входной файл состоит из $N+1$ строки ($0 < N < 700$). В первой строке записано число N — количество населенных пунктов на карте. В каждой следующей строке через пробел записано по два целых числа — координаты населенного пункта на карте, все координаты по модулю не превосходят 10^5 .

Выходные данные

В выходном файле должно быть одно целое число — максимальное количество населенных пунктов, через которые может проехать Емеля на своей печке.

Пример

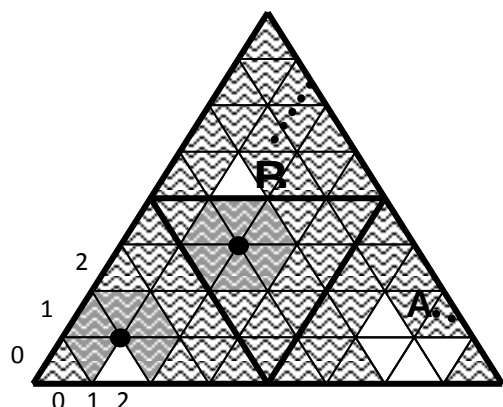
<i>input.txt</i>	<i>output.txt</i>
13 0 0 3 3 200 201 11 5 19 0 8 200 8 8 0 13 200 7 5 13 14 13 8 20 18 13	4
3 1 1 3 3 -1 -1	3

Задача 6. Третий мир

Ограничение по времени: 1 секунда на тест

На тетраэдрической планете в треугольных землях живут существа с 9 пальцами (по три на каждой руке). Сейчас, после создания кораблей, у них возникла проблема поиска оптимального пути между двумя морскими портами. Из-за геометрии их планеты, карты у них треугольные и являются разверткой тетраэдра. Одна из таких карт показана на рисунке, в ней ребра тетраэдра нарисованы жирными линиями, а суша обозначена белыми треугольниками. Остальное место

занимает вода, по которой может проходить путь. Порты располагаются на воде и обязательно примыкают к суше.



Существуют также рифы, обозначенные на карте черными точками, и находиться они могут только в вершинах треугольников. По треугольникам, прилегающим к рифам, путь проходить не может. Известно, что порты не прилегают к рифам.

Система координат у существ следующая: для любого треугольника его первой координатой является число треугольников, расположенных слева от него, а второй – номер ряда, в котором находится данный треугольник. Нумерация рядов производится снизу вверх и начинается с нуля. Риф задается координатами треугольника, в вершине которого он находится. Если треугольник направлен вверх, то риф в верхней вершине, если вниз — то в нижней.

Длиной пути называется число треугольников, по которым проходит путь, включая начальный и конечный треугольники. Путь должен пересекать только стороны соседних треугольников, и не может проходить через вершины. Ваша задача: написать программу, которая определяет длину кратчайшего пути между двумя заданными портами. Один из кратчайших путей между портами A и B показан на карте пунктирной линией.

Входные данные

В первой строке входного файла записано число N ($1 \leq N \leq 50$) — длина ребра тетраэдра. Во второй — две пары чисел — координаты портов. Третья строка содержит число K — количество клеток суши. В следующие K строк записано по два числа на строку — координаты клеток суши. Далее идет строка, в которой задается количество рифов L , а затем L строк описывают координаты рифов.

Выходные данные

В выходной файл нужно выдать целое число — длину кратчайшего пути из одного порта в другой. Если такого пути не существует, или длина пути больше 10^3 , то выдать 0.

!!! Все входные и выходные данные представлены в троичной системе счисления.

Пример

input.txt	output.txt
<pre> 11 102 1 10 11 20 2 0 101 1 102 0 110 0 111 0 2 11 2 2 0 10 10 </pre>	<pre> 20 </pre>

Задача 7. Камни

Ограничение по времени: 1 секунда на тест

Гора состоит из площадок, соединенных между собой узкими проходами. На каждой площадке лежит какое-то количество камней. У разных площадок разный рейтинг, зависящий от высоты площадки — чем выше площадка, тем больше ее рейтинг среди камней. Очевидно, что на площадках с более высоким рейтингом лежит большее количество камней.

В один солнечный день главный камень одной из площадок обратил внимание, что вокруг стало очень тесно. Но камни не могут катиться вверх, а на нижние площадки они катиться не хотят из-за плохого рейтинга. Поэтому главный камень пошел на хитрость. Он выбрал M камней и сказал каждому из них, что где-то внизу появилась замечательная площадка с самым высоким рейтингом на горе. Для полного правдоподобия он составил карту площадок, на которые можно скатиться с его площадки. Всего, включая переполненную площадку, оказалось N площадок. Поскольку главный камень ленивый, он не стал рисовать все пути между площадками, а просто нарисовал минимальное количество проходов между площадками, чтобы из его площадки можно было добраться в любую, находящуюся ниже. Он показал каждому из M камней эту карту и объяснил, где находится замечательная площадка (чтобы камень запомнил путь). Причем, чтобы главные камни нижних площадок не имели к нему претензий из-за такого нашествия, некоторым камням он показал на другие площадки, чем остальным. Чтобы не создать лавину, он сказал камням катиться с небольшим интервалом между собой. Немного подумав, он решил, что можно пускать камни парами, чтобы пространство быстрее освободилось. Камни ему поверили и стали собираться в дорогу.

Камни не любят скучать, поэтому каждая пара решила посчитать, сколько времени они будут катиться вместе, и, может быть, перестроить пары, чтобы увеличить это время. На то, чтобы скатиться по проходу с одной площадки в следующую, не заходя по пути на другие площадки, камень тратит одну минуту. Но складывать числа камни не умеют, в этом им требуется ваша помощь.

Входные данные

В первой строке входного файла записаны два целых числа — N и K ($1 \leq N \leq 25000$, $K = \frac{M}{2}$, $0 \leq M \leq 2000$). В следующих $N - 1$ строках находятся пары чисел i и j , означающие, что из i -й площадки можно напрямую скатиться на j -ю. Площадки занумерованы числами от 1 до N , верхняя площадка, откуда катятся камни, может иметь любой номер. Записанных пар номеров площадок достаточно, чтобы определить путь из самой верхней площадки до любой другой. Дальше идут K строк, на каждой строке через пробел записана пара чисел a и b ($1 \leq a, b \leq N$), обозначающая пункты назначения для очередной пары камней: первый камень катится на площадку a , а второй — на площадку b .

Выходные данные

В выходной файл нужно вывести K строк, на каждой строке должно находиться по три целых числа a , b и c , записанных через пробел. Эти числа означают, что пара камней, направляющихся на площадки a и b , будет катиться вместе с минут. Тройки чисел нужно выдавать в том же порядке, в котором перечислены соответствующие им пары во входном файле.

Пример

input.txt	output.txt
6 2	2 4 0
1 2	5 6 1
1 3	
1 4	
3 5	
3 6	
2 4	
5 6	

Задача 8. Долина мертвых

Ограничение по времени: 1 секунда на тест

Гондор — один из последних оплотов сил, противостоящих Мордору. Гондору угрожает серьёзная опасность. Орки продолжают свои набеги, и силы защитников тают. Чтобы спасти положение Арагорн, потомственный король Гондора, отправляется в долину мертвых, чтобы призвать души мертвых выполнить их долг перед королями Гондора. Легенды гласят, что никто живым не возвращался оттуда. На входе Арагорн увидел руны, гласящие, что мертвые при помощи магии научились делать хроноколодцы, позволяющие им путешествовать во времени.

Хроноколодец — это образование, которое имеет фиксированные пространственные координаты и осуществляет перенос по некоторому промежутку времени. Любой объект, попавший в хроноколодец, переносится во времени к его началу. Начало колодца может быть либо ниже во времени конца — тогда он действует в прошлое, либо выше — тогда он действует в будущее. Однако проклятье не даёт им покинуть долину. Несмотря на возможность путешествий во времени, мертвые предпочитают жить в K -ом году по летоисчислению Гондора. Там же была приведена схема расположения хроноколодцев. Хроноколодцами могут воспользоваться и простые люди, вот только не каждый сможет пройти обратно тем же путём. Если на пути Арагорна встречается хроноколодец, то он его может обойти без увеличения длины пути. Поскольку долина большая, Арагорну хочется узнать, какое минимальное расстояние он пройдёт, прежде чем сумеет найти мертвых. Помогите ему в этом.

Входные данные

В первой строке входного файла записано число N — количество хроноколодцев ($1 \leq N \leq 5000$). Во второй строке через пробел записаны два числа: текущий год по летосчислению Гондора M и год назначения K . В третьей строке записаны координаты входа в долину. Если место находится внутри хроноколодца, то Арагорн немедленно перемещается в его начало. В следующих N строках записаны четвёрки целых чисел $X_i, Y_i, Start_i, End_i$ — координаты i -го хроноколодца, его начало и конец во времени ($1 \leq i \leq N$). Все координаты даны в метрах — это целые числа, по модулю не превосходящие 10^4 , все времена — целые числа, по модулю не превосходящие 10^9 . Гарантируется, что никакие два хроноколодца не имеют общих точек во времени.

Выходные данные

В выходном файле должно стоять единственное число — суммарная длина пройденного Арагорном пути с точностью до 0.01. Если такого пути не существует, нужно вывести -1 .

Примеры

<i>input.txt</i>	<i>output.txt</i>
4 10 -1 0 0 1 1 -1 11 1 -1 -1 11 -1 -1 -1 11 -1 1 -1 11	1.41
3 0 10 0 5 -2 5 -10 0 2 5 20 0 0 0 10 4	-1

Задача 9. Шахматы

Ограничение по времени: 1 секунда на тест

Таня играла с папой в шахматы. Она играла белыми фигурами, а папа — черными. В какой-то момент, когда на поле сложилась такая ситуация, что у белых имелись ладья и король, а у черных — только король, по телевизору начался футбол, и папа, сказав, что сдается, пошел смотреть телевизор. Однако Таню заинтересовало, какое минимальное число ходов ей потребовалось бы, чтобы выиграть у папы, при условии, что они оба играли бы по оптимальной стратегии. Ваша задача состоит в том, чтобы определить искомое минимальное количество ходов.

В игре "шахматы" игроки ходят по очереди одной из своих фигур, в нашем случае белые ходят королем или ладьей, а черные — только королем. Фигуры располагаются в клетках квадратного поля размером 8×8 .

- Король может ходить на любую из клеток, граничащих с его текущей клеткой, по прямой или по диагонали, за исключением клеток, на которых стоит либо он сам, либо ладья того же цвета.
- Ладья может ходить на любую клетку, находящуюся на той же вертикали или горизонтали, что и ее текущая клетка, за исключением клеток, на которых стоит либо она сама, либо король того же цвета. При этом все клетки от исходной позиции до новой должны быть свободны.
- Король не может ходить на ту клетку, которая находится "под боем" фигуры его противника, то есть на ту клетку, на которую может сходить следующим ходом одна из фигур его противника. Также, если король оказался "под боем", то этим ходом он должен сходить на клетку, которая не находится "под боем".
- На одной и той же клетке не могут находиться одновременно две или больше фигур. Если с очередным ходом фигура становится на одну клетку с фигурой противника, то фигура противника удаляется с поля и больше не участвует в игре.
- Если игроку подошла очередь ходить, но ходить ему некуда, то игра заканчивается. При этом если его король в этой ситуации находится "под боем", то этот игрок проигрывает, а если нет — то игра заканчивается вничью.

В отличие от профессиональных шахмат, Таня с папой играют в упрощенные шахматы, в которых не действует правило, согласно которому при трехкратном повторении какой-либо позиции игра заканчивается вничью. Иными словами, позиции в процессе игры могут повторяться сколько угодно раз.

Оптимальность стратегии заключается в том, что игрок при каждом своем ходе выбирает тот ход, при котором он может выиграть при любой игре противника. При этом если таких ходов несколько, то выбирается тот ход, при котором выигрыш достигается за минимальное число ходов. Если выигрышных ходов нет, то выбирается ход, ведущий к ничьей. Если и таких ходов нет, то выбирается ход, при котором игрок проигрывает за максимальное число ходов.

Входные данные

Первые три строки входного файла содержат описания позиций белого короля, белой ладьи и черного короля, соответственно, по одному на каждой строке. Позиция каждой фигуры задается двумя подряд идущими символами. Первый символ является заглавной буквой латинского алфавита, A, B, C, D, E, F, G или H и обозначает номер вертикали клетки, на которой стоит фигура (A — первая вертикаль, B — вторая, ..., H — восьмая). Второй символ — цифра от 1 до 8, которая равна номеру горизонтали клетки.

В четвертой строке записана одна из двух латинских букв — W, если в заданной позиции ходят белые, или B, если ходят черные.

Выходные данные

В единственную строку выходного файла требуется вывести одно число, равное минимальному количеству ходов, за которые белые могут выиграть у черных. При этом в количество ходов включаются как ходы белых, так и ходы черных. Если исходная ситуация выигрышна для белых, то нужно вывести 0. Если она ничейная, или белые не могут выиграть за конечное число ходов при оптимальной игре обоих игроков, то следует вывести число -1.

Пример

<i>input.txt</i>	<i>output.txt</i>
C6 G7 B8 W	5

Задача 10. Суперсвадьба

Ограничение по времени: 1 секунда на тест

Съемки сериала «Любовь любителей любви» подходят к концу, и сценаристы решили поместить в конец сериала суперсвадьбу. Под суперсвадьбой понимается одновременное проведение N свадеб между N мужчинами и N женщинами. Для каждого мужчины и каждой женщины известно, могут ли они пожениться. Продюсера сериала заинтересовал вопрос: чётно ли число возможных суперсвадеб? Напишите программу, которая ответит на этот вопрос.

Входные данные

В первой строке входного файла записано одно целое число — количество тестов, не более 10. Описание каждого теста состоит из нескольких строк. Сначала в отдельной строке входного файла записано число целое N ($1 \leq N \leq 100$). В каждой из следующих N строк через пробел записано по N чисел. Эти числа задают таблицу размера $N \times N$, где j -ое число в i -й строке равно 1 тогда и только тогда, когда мужчина номер i и женщина номер j могут пожениться в конце сериала, в противном случае на этом месте стоит 0.

Выходные данные

В выходной файл для каждого теста в отдельной строке нужно вывести **EVEN**, если количество возможных суперсвадеб чётно, и **ODD** — в противном случае.

Пример

<i>input.txt</i>	<i>output.txt</i>
1 3 1 0 0 1 1 1 1 0 1	ODD