Задача 1. Расстановка чисел

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	128 мегабайт

Недавно Вася решал на олимпиаде по математике задачу о расстановке чисел от 1 до 37 в ряд в таком порядке, что каждое число, начиная со второго по 37-е, делит сумму всех чисел, стоящих левее него. Оказалось, что если первое число равно 37, то третье всегда будет 2! Поскольку Вася – начинающий программист, он задумался, всегда ли можно расставить числа от 1 до n в ряд таким образом? Сам он решить эту задачу так и не смог, поэтому просит Вас помочь написать программу, которая по числу n либо будет предъявлять требуемую расстановку чисел в ряд, либо будет сообщать, что такой расстановки не существует.

Формат входных данных

В первой строке входного файла содержится единственное целое число n ($2 \leqslant n \leqslant 10000$).

Формат выходных данных

Если можно выписать числа от 1 до n в ряд, указанным выше способом, то в первую строку выходного файла нужно вывести n чисел через пробел – любую требуемую расстановку этих чисел. Если такой расстановки не существует, то выведите в первую строку выходного файла единственное число -1.

Примеры

input.txt	output.txt
2	2 1
3	2 1 3

Задача 2. Дед Мороз

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	4 секунды
Ограничение по памяти:	128 мегабайт

Дед Мороз решил зайти в магазин за подарками. Магазин состоит из отделов, соединённых между собой коридорами. Каждый коридор по-праздничному украшен дизайнерами в один из двух цветов – белый или красный. Известно, что для любого отдела число выходящих из него коридоров красного цвета равно числу коридоров белого цвета.

В каждом коридоре можно купить уникальные сувениры, поэтому Дед Мороз хочет посетить все коридоры. В то же время Дед Мороз жутко не любит однообразие, поэтому хочет посетить каждый коридор ровно один раз, при этом так, чтобы цвета коридоров обязательно чередовались. Например, если Дед Мороз зайдёт в отдел через красный коридор, то в следующий отдел он может пойти только через белый коридор. В любой отдел Дед Мороз может заходить несколько раз.

Гарантируется, что любые два отдела соединяет не более, чем один коридор. По каждому коридору можно проходить в обе стороны. Помогите Деду Морозу найти путь в магазине, удовлетворяющий условиям выше, чтобы в конце он вернулся в тот же отдел, с которого начинался обход.

Формат входных данных

В первой строке входного файла даны два целых числа N и M – число отделов в магазине и число коридоров ($4 \leqslant N, M \leqslant 300000$).

В каждой из следующих M строк содержится информация о коридорах. Каждая строка содержит два целых чисел a_i, b_i – номера отделов, которые соединяет i-ый коридор, и его цвет c_i ($1 \leqslant a_i, b_i \leqslant N$). Цвет задаётся словом «red» или «white» (без ковычек). Предполагается, что Дед Мороз хочет начать обход с отдела, имеющего номер 1.

Формат выходных данных

Требуется вывести $M + 1$ число через пробел – номера отделов на протяжении пути Деда Мороза, которые он будет посещать, проходя по коридорам. Гарантируется, что путь существует.

Примеры

input.txt	output.txt
4 4 1 2 white 2 3 red 3 4 white 4 1 red	1 2 3 4 1
7 8 1 2 white 2 3 red 3 4 white 4 1 red 7 1 white 5 6 white 1 5 red 6 7 red	1 7 6 5 1 2 3 4 1

Задача 3. Язык JTC

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	3 секунды
Ограничение по памяти:	128 мегабайт

С каждым годом появляется всё больше и больше различных технологий и инструментов, в том числе и для программистов, но появление языка Just Type Code (JTC) – это настоящий прорыв в области разработки программного обеспечения. Конечно, всё и сразу продумать сложно, поэтому пока в JTC есть всего три операции:

- Оператор задания переменной нового значения. Этот оператор записывается как **new a**, где **a** – название переменной. Самое удобное здесь, что разработчику даже нет необходимости указывать значение переменной – при выполнении этого оператора значение переменной может стать любым в пределах числового типа JTC. Для каждой переменной этот оператор может встречаться в программе не более одного раза.

- Оператор уточнения значения переменной. Оператор записывается как **state x** $<$ **y**, где **x** и **y** могут быть некоторыми уже определёнными ранее переменными или константами в пределах числового типа JTC. Этот оператор указывает, что выполняется неравенство **x** $<$ **y**. Оператор не должен противоречить предыдущим операторам **state**.

- Оператор сравнения, имеющий вид **ask x** $<$ **y**, где **x** и **y** могут быть некоторыми уже определёнными ранее переменными или константами в пределах числового типа JTC. Для этого оператора программа тут же выводит на экран одно из следующих слов, на основе ранее введённых фактов:

  «sure», если обязательно выполняется **x** $<$ **y**;

  «possible», если условие **x** $<$ **y** может как выполняться, так и не выполняться;

  «impossible», если условие **x** $<$ **y** точно не выполняется.

Так как многие разработчики жаловались, что в языках программирования прошлого поколения границы типов данных были сложными для запоминания, в JTC все константы и переменные всегда лежат в диапазоне от -10^9 до 10^9 включительно. Каждая переменная в JTC описывается одной строчной буквой латинского алфавита.

Чтобы использовать исключительно передовые технологии в области разработки ПО, вам нужно написать интерпретатор языка программирования JTC.

Формат входных данных

В первой строке входного файла дано одно число N – количество операций в программе на JTC, которую требуется выполнить ($1 \leqslant N \leqslant 10^5$).

Каждая из следующих N строк содержит ровно один оператор программы на языке JTC в описанном выше формате. Гарантируется, что программа корректна – переменные не используются до задания им некоторого значения и факты **state** непротиворечивы. Операции заданы в порядке их выполнения.

Формат выходных данных

Для каждой операции **ask** необходимо вывести слово «sure», «possible» или «impossible» в отдельную строку, в зависимости от результата сравнения.

Пример

input.txt	output.txt
6 new a state a < 1 ask a < 1 ask 0 < a new b ask a < b	sure impossible possible

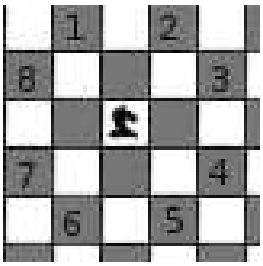
Задача 4. Лошадью ходи...

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	128 мегабайта

Вася и Аня – любители игры шахматы. В последнее время они играют в шахматы «вслепую»: когда игроки во время игры не видят доски. Для тренировки они придумали следующую игру с одним конём на доске. Сначала Аня загадывает размер шахматной доски, а также положение коня на этой доске. Затем Вася пытается довести коня до верхней левой клетки доски, вслепую управляя конём. Игра разбивается на шаги, на каждом из которых Вася делает ход конём, либо говорит, что останавливает игру, если уверен, что конь находится в верхней левой клетке доски.

Вам предлагается написать программу, которая будет играть в эту игру за Васю. У вас нет права на ошибку: только один раз можно сообщить о том, что конь находится в требуемой клетке.

Как известно, всего есть не более 8 различных способов походить из данной клетки конём: либо сдвинуть коня на две клетки влево или вправо, а потом на одну вверх или вниз, либо сдвинуть коня на две клетки вниз или вверх, а потом на одну клетку влево или вправо. Возможные ходы приведены на следующем рисунке.



Все эти ходы можно описать в следующем формате: «m_1 n_1 m_2 n_2» (без ковычек) – где m_1 и m_2 – это символы из множества $\{l, r, u, d\}$, отвечающие за направление хода, а n_1 и n_2 – количество клеток, на которое нужно переместить коня в соответствующем направлении. Буква l означает, что коня нужно двигать налево, r – направо, u – вверх и d – вниз. Числа n_1 и n_2 всегда равны 1 или 2 каждое. Например, ход на рисунке, отмеченный номером 3, можно записать либо как «r 2 u 1», либо как «u 1 r 2», поскольку, чтобы сделать ход, нужно сдвинуться на две клетки вправо и на одну вверх.

Каждый раз программа должна сообщать ход в описанном формате, либо сообщать, что конь находится в требуемой угловой клетке. На каждый свой ход программа получает один из следующих двух ответов:

- `continue` — если данный ход сделать можно, при этом конь перемещается согласно ходу в новую клетку;
- `out of board` — данный ход сделать нельзя, поскольку после этого хода конь окажется за доской, при этом конь остается на прежнем месте;

Если игрок уверен, что конь находится в верхнем левом углу доски, то вместо хода он должен вывести «stop». Если конь действительно находится в нужной клетке, то игрок победил, иначе он проигрывает и игра завершается.

Игровое поле представляет собой квадрат $N \times N$ клеток. Согласно правилам игры, Аня может установить значение N в диапазоне от 4 до 100 включительно. В одной из N^2 клеток доски она размещает коня.

Протокол взаимодействия

Это интерактивная задача, и в ней вам предстоит работать не с файловым вводом-выводом, а со специальной программой — интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

Ваша программа выводит на поток вывода ходы для коня. Каждая команда задаётся строкой «m_1 n_1 m_2 n_1» (см. описание команд выше). После каждого выведенного вашей программой хода на поток ввода приходит один из ответов: `continue` или `out of board` (см. описание результатов выше).

После вывода команды `stop` игра заканчивается. Если конь не находится в верхней левой угловой клетке доски, то ваше решение получает вердикт Wrong Answer. Количество команд не должно превышать $20N^2$, иначе решение получит вердикт Wrong Answer.

Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда `flush` языка) после каждой выведенной команды. Иначе решение может получить вердикт Timeout.

Пример

стандартный ввод	стандартный вывод
r 2 u 1 l 2 u 1 l 2 u 1 stop	continue continue out of board

Задача 5. Замена чисел

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	3 секунды
Ограничение по памяти:	128 мегабайт

Сегодняшний урок математики очень заинтересовал Ванечку – на уроке проходили делимость чисел. Учитель предложил детям следующее задание: пусть имеется натуральное число n. Если $n = ab$ для некоторых натуральных чисел a, b, то число n можно заменить на число $a+b-2$. После этого по этому же правилу можно заменять новое число, рассматривая его вместо числа n и т.д. Оказалось, что с какого бы числа дети не начинали замены, рано или поздно у них всегда получалось число 1. Ванечка уже понял почему у них так получалось, но теперь он задумался за какое минимальное число замен можно из данного числа получить число 1? Помогите Ванечке написать программу, которая поможет ему экспериментировать с новой задачей.

Формат входных данных

В первой строке входного файла содержится число n ($1 \leqslant n \leqslant 10^9$).

Формат выходных данных

В выходной файл нужно вывести единственное число – минимально количество замен, которые позволяют получить число 1, если исходное число равно n.

Примеры

input.txt	output.txt
2	1
6	3

Замечание

В первом примере, поскольку $2 = 1 \cdot 2$, то число 2 сразу можно заменить на число $1 + 2 - 2 = 1$. Во втором примере из числа 6 после первой замены можно получить числа 3 или 5. После двух замен получаются числа 2 или 4. Наконец, после третьей замены можно получить число 1 и поэтому наименьшее число замен равно трём.

Задача 6. Схема тоннелей

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	128 мегабайт

Король Берляндии очень любит инновации. В этом году он задумал соединить города новым видом транспорта под названием «Пупер-луп». Для того чтобы запустить новый транспорт, нужно прокопать тоннели между некоторыми городами. Король хочет, чтобы жители всех городов страны были довольны, поэтому решено прокопать тоннели так, чтобы из каждого города выходило одинаковое число тоннелей. При этом дворец короля располагается за пределами всех городов, поэтому один тоннель будет идти от дворца короля до некоторого города. Каждый тоннель соединяет либо два города, либо дворец короля Берляндии и некоторый город. Естественно, король хочет, чтобы с помощью нового транспорта можно было добраться из любого города до любого другого. Помогите королю понять, можно ли соединить города с помощью тоннелей требуемым образом?

Формат входных данных

В первой строке входного файла содержатся два целых числа n и k – число городов в Берляндии и число тоннелей, которое должно выходить из всех городов ($1 \leqslant n \leqslant 500$, $1 \leqslant k \leqslant n$).

Формат выходных данных

Если можно придумать требуемую схему тоннелей, то в первой строке выходного файла выведите «YES». Следующие строки должны описывать тоннели. Для каждого тоннеля выведите пару чисел a и b, разделённую символом пробела, где a и b – номера городов или дворца, между которыми нужно будет прокопать тоннель. Предполагается, что города имеют номера $1, 2, ..., n$. Для обозначения дворца используйте число 0. Информацию про каждый тоннель нужно выводить ровно один раз.

Если схему тоннелей составить нельзя, то выведите единственное слово «NO».

Примеры

input.txt	output.txt
4 2	NO
5 3	YES 0 1 1 3 1 2 4 5 4 2 4 3 5 2 5 3

Задача 7. Домашнее задание

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	128 мегабайт

Паша посещает спецкурс по быстропечатанию на клавиатуре. Он уже подготовил домашнее задание – каждый слушатель спецкурса должен придумать к следующему занятию строку, которую будут набирать другие участники. Однако перед отправкой строки по электронной почте преподавателю он обнаружил, что не до конца прочитал письмо с заданием. В конце письма написано, что для тренировки двух букв все слова в строке должны состоять из одних и тех же двух букв!

Помогите Паше срочно найти наименьшее число символов строки, которые надо заменить, чтобы новая строка уже подходила под условие домашнего задания.

Формат входных данных

В первой строке входного файла содержится строка, которую подготовил Паша. Все символы этой строки – это либо латинские строчные буквы, либо знак пробела. Гарантируется, что строка содержит хотя бы две буквы и состоит из не более, чем 10^4 символов.

Формат выходных данных

В выходной файл выведите единственное число – наименьшее количество букв, которые нужно заменить в строке Паши, чтобы она состояла только из двух разных латинских букв. Символы пробела заменять не нужно.

Примеры

input.txt	output.txt
mama	0
ded moroz	4
aa aaa aaaaa	1

Задача 8. Подсчет молекул

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	128 мегабайт

Начинающий физик Женя изучает газы в лаборатории элементарных частиц. Как самому ответственному студенту, ему предложили решить очень интересную задачу: в кубическом сосуде со стороной N сантиметров находится газ. На этот газ воздействуют совершенно новой и совершенно секретной установкой. Эта установка позволяет остановить броуновское движение частиц и продержать их в таком «замороженном» состоянии некоторое время.

Установка тратит очень много энергии на поддержание такого состояния во всем сосуде, поэтому ученые хотят найти в нем такую область в форме параллелепипеда, в которой сосредоточено наибольшее количество молекул газа. Задача Жени – разработать программу, которая внутри сосуда находит такую область, чтобы ученые могли запустить следующий эксперимент.

Предполагается, что сосуд поделен на единичные кубические сантиметры от одного из своих углов. Требуемая область должна содержать целое число кубических сантиметров, то есть никакой кубический сантиметр не может содержаться в области частично. Женя обратился за помощью к вам, как к квалифицированному специалисту в области программирования.

Формат входных данных

В первой строке входного файла содержится четыре целых числа: N, a, b, c – размер кубического сосуда с газом и длина, ширина, высота искомой области ($1 \leqslant N \leqslant 100$, $1 \leqslant a, b, c \leqslant N$). Во второй строке содержится число M – количество молекул газа в сосуде ($1 \leqslant M \leqslant 10^5$). Известно, что в каждом кубическом сантиметре сосуда может содержаться больше, чем одна молекула газа.

В следующих N строках содержится описание положения молекул в сосуде. Каждая строка содержит три числа x, y, z – координаты кубического сантиметра, в котором содержится очередная молекула ($1 \leqslant x, y, z \leqslant N$). Точкой отсчета считается левый верхний дальний угол сосуда, кубические сантиметры нумеруются от него по длине, ширине и высоте.

Формат выходных данных

В выходной файл необходимо вывести ответ на задачу – максимальное количество молекул газа, которое может оказаться в параллелепипеде.

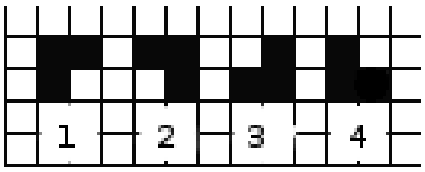
Примеры

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

Задача 9. Замощение уголками

Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	128 мегабайт

Понятно, что из квадрата 2×2, составленного из четырёх единичных квадратов, удаляя один квадрат, можно получить четыре разных типа фигур, составленных из трёх квадратов. Будем для краткости называть эти фигуры *уголками*. Мы нумеруем эти четыре типа, как показано на следующем рисунке.



Предположим, что на клетчатом поле фиксирован прямоугольник $n \times m$, и из этого прямоугольника удалена одна клетка. Рассмотрим все разбиения полученной фигуры на уголки. Даны стоимости использования каждого из 4-х видов уголков. Нужно посчитать количество разбиений минимальной стоимости.

Формат входных данных

Первая строка входного файла содержит четыре числа n, m, x, y, где n – число строк у прямоугольника, m – число столбцов, а x, y – номера строки и столбца выкинутой клетки (нумерация идет по строкам от верхней к нижней, по столбцам от левого к правому) ($1 \leqslant n, m$, $n + m < 20$, $1 \leqslant x \leqslant n$, $1 \leqslant x \leqslant m$). Вторая строка содержит четыре целых числа c_1, c_2, c_3, c_4 – стоимости использования каждого вида уголков ($1 \leqslant c_1, c_2, c_3, c_4 \leqslant 10$).

Формат выходных данных

В выходной файл требуется вывести два целых числа через пробел – минимальную стоимость замощения доски без клетки и количество способов это сделать. Гарантируется, что хотя бы одно разбиение доски существует.

Пример

input.txt	output.txt
4 4 1 1 1 1 1 1	5 1

Замечание

В примере для квадрата 4×4 без левой верхней клетки существует единственное разбиение на уголки из трех квадратов.