Задача A. Казино в домике

Имя входного файла:	`game.in`
Имя выходного файла:	`game.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

Одно широко известное казино, находящееся в не совсем скучном саду, поменяло специализацию. Сейчас там вместо рулетки играют в командную орлянку. Всего в игре не более $2N - 1$ раундов. В каждом раунде бросается монета. Если выпадает «орёл», то очко решением крупье идёт команде гостей, если «решка» — то команде хозяев. Команда, набравшая N очков первой, объявляется победителем и забирает банк.

Однажды некто Дозоров, один из завсегдатаев казино, случайно задел какой-то провод, после чего свет в помещении казино вырубился, и все радостно стали бить Дозорова. Крупье принял решение прервать игру и разделить банк в соответствии с шансами каждой из команд на выигрыш. Ваша задача — по заданному числу N, счёту в партии на момент остановки игры и сумме в банке определить, какую сумму получит каждая сторона.

Формат входного файла

В первой строке входного файла задано $T \leq 40$ — количество тестовых примеров. Каждая из T следующих строк содержит 4 целых числа: N — количество набранных очков, требуемое для победы ($1 \leq N \leq 50$), k_1, k_2 — количество очков, набранное командами хозяев и гостей соответственно ($0 \leq k_1, k_2 \leq N$), S — сумма, находящаяся в банке ($0 \leq S \leq 10^{32}$). При этом гарантируется, что сумму можно разделить нацело в соответствии с требованием задачи. Вероятности выпадения «орла» и «решки» считать одинаковыми и равными 0.5.

Формат выходного файла

В выходной файл выведите два целых числа, в сумме дающих S — суммы денег, которые получат соответственно команда хозяев и команда гостей.

Пример

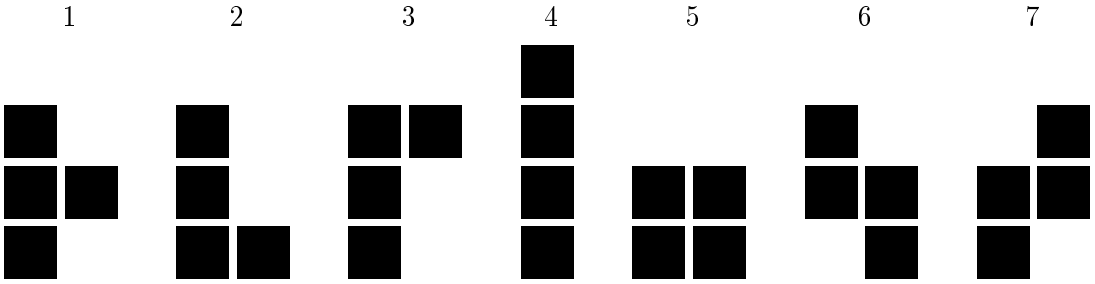
game.in	game.out
3 3 1 2 16 4 1 1 2 5 1 4 32	4 12 1 1 2 30

Задача B. Игра «Сиртет»

Имя входного файла:	`tetris.in`
Имя выходного файла:	`tetris.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

Всем известная игра «Тетрис» заключается в заполнении прямоугольного стакана фигурками, занимающими 4 клетки. Когда в стакане образуется ряд, полностью заполненный клетками, принадлежащими фигуркам, этот ряд убирается, а все ряды, находившиеся над ним, смещаются на одну позицию вниз. Одновременно могут быть убраны несколько рядов.

Все возможные фигурки тетриса и их начальные ориентации приведены ниже.



Фирма OrcimSoft 1 апреля выпустила новую игру — «Сиртет». В этой игре в прямоугольный стакан падают стандартные фигурки тетриса, раскрашенные в 7 цветов, причём каждая клетка падающей фигурки раскрашена в свой цвет. Когда в заполненной части стакана образуется комбинация клеток одного цвета, образующая фигурку тетриса, эти клетки удаляются, а заполненные клетки, находившиеся над ними, сдвигаются вниз. Цель игры — удалить наибольшее количество клеток.

Когда в верхней части стакана появляется очередная фигурка, игрок может сдвинуть её по горизонтали или повернуть, после этого он «сбрасывает» фигурку вниз, причём он уже не может сдвигать её по горизонтали или поворачивать. Фигурка падает вниз до тех пор, пока дальнейшее движение не станет невозможным. Затем удаляются все образовавшиеся одноцветные фигурки. Если существует несколько вариантов удаления фигурок, действуют правила, описанные ниже. Все фигурки удаляются одновременно, после чего заполненные клетки, находившиеся над удалёнными, сдвигаются на столько рядов вниз, сколько клеток было удалено в соответствующем столбце. Если образовались новые одноцветные комбинации, они удаляются, клетки над ними смещаются вниз и т. д.

Правила удаления фигурок. *Опорной клеткой* фигурки в некоторой ориентации назовём самую нижнюю из самых левых клеток фигурки в этой ориентации. В стакане ищется самая левая из самых нижних опорных клеток одноцветных фигурок. Если найденная опорная клетка может принадлежать сразу нескольким одноцветным фигуркам в различной ориентации, выбирается фигурка с наименьшим номером (номер фигурки показан на рисунке выше), а если после этого неоднозначность не устраняется, выбирается фигурка с наименьшим углом поворота против часовой стрелки относительно начальной ориентации (см. выше), причём угол поворота измеряется в прямых углах от 0 до 3, то есть 0 означает, что ориентация фигурки совпадает с начальной, а 3 — что фигурка повёрнута против часовой стрелки на 270 градусов относительно начальной ориентации. Затем согласно этим же правилам ищутся опорные клетки других фигурок, причём клетки, принадлежащие ранее найденным фигуркам, не могут использоваться как части новых фигурок.

Стакан имеет ширину 10 клеток и высоту 40 клеток. Стакан в начале игры пуст. Ваша задача состоит в том, чтобы промоделировать игру, то есть получить финальную конфигурацию стакана при заданной последовательности бросаний фигурок.

Формат входного файла

Во входном файле перечисляется последовательность ходов, сделанных игроком. Каждый ход записывается на отдельной строке. В начале идут пять чисел, задающих выпавшую фигурку: сначала её номер (целое число от 1 до 7), затем цвета четырех её клеток, причем клетки перечисляются

снизу вверх и слева направо согласно начальной ориентации фигурки (см. выше). Цвет задаётся целым числом от 1 до 7. Затем парой (колонка, ориентация) задаётся ход игрока. Ориентация (**целое число, по модулю не превышающее 30000**) — это угол поворота фигурки против часовой стрелки, измеренный в прямых углах. Колонка — это номер колонки, в которой находится опорная клетка фигурки в данной ориентации, считая слева направо от 1 до 10. Гарантируется, что в верхней части стакана всегда останется как минимум 5 свободных рядов, и что фигурка в заданной ориентации с заданной опорной клеткой помещается в стакане (не выходит за правую границу).

Формат выходного файла

Напечатайте конфигурацию стакана. Если клетка пуста, в этом месте печатайте символ `.` (точка), а в противном случае напечатайте цвет клетки (цифра от 1 до 7). Незаполненные ряды в верхней части стакана не печатаются. Если стакан пуст, напечатайте единственную строку `EMPTY`.

Пример

tetris.in	tetris.out
1 3 2 1 2 1 1 7 2 2 2 4 1 0	2......... 143.......

Задача C. Ваша жаба, вы и кормите

Имя входного файла:	`frog.in`
Имя выходного файла:	`frog.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

В целях борьбы с кризисом компания «Jedi Games» выпустила новую игру для топ-менеджеров «Накорми свою жабу». Итак, в каждой клетке квадратного игрового поля $N \times N$ клеток находится комар весом a_{ij}, где i — номер строки, j — номер столбца. Жаба, прыгая с клетки на клетку, в некоторых их них ест комаров. Правила игры таковы: в каждом столбце можно съесть не более одного комара. Всякий раз при съедании комара запоминаем номер строки, откуда был съеден комар, и сумма номеров строк, в которых были съедены комары, в конце игры должна быть в точности равна N. Учтите, что если из какой-то строки съедено несколько комаров, то номер данной строки участвует в суммировании более одного раза. Определите максимальный суммарный вес съеденных комаров при следовании приведённым правилам.

Формат входного файла

Первая строка входного файла содержит количество тестов $1 \le T \le 10$. Первая строка каждого теста содержит число N ($1 \le N \le 50$). Следующие N строк содержат по N целых чисел a_{ij} ($0 \le a_{ij} \le 50$), разделённых пробелами.

Формат выходного файла

Для каждого теста в выходной файл выводится одно целое число — вес съеденных комаров.

Пример

frog.in	frog.out
2 3 8 2 1 1 2 6 2 7 2 5 8 2 1 2 3 1 2 6 2 4 2 7 2 3 4 1 3 2 4 4 1 3 4 3 1	14 19

Задача D. Проблема Билла

Имя входного файла:	`bill.in`
Имя выходного файла:	`bill.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

На праздновании очередного этапа карьерного роста своей жены известный саксофонист Билл планирует исполнить композицию “My Little Trick” группы “Strait Dires”. Однако за час до выступления оказалось, что его секретарша Моника перепутала ноты, дав ему ту же мелодию, но для партии балалайки, оказавшуюся в другой тональности.

Напишите программу, которая поможет Биллу восстановить исходную партитуру с помощью транспонирования нот из одной тональности в другую.

В нотной записи используется 12 нот, обозначаемых заглавными буквами от A до G, за которыми, возможно, следует знак диеза `'#'` или бемоля `b`. Ноты упорядочены циклически, как показано ниже. Косая черта разделяет различные обозначения одной и той же ноты.
C/B# C#/Db D D#/Eb E/Fb F/E# F#/Gb G G#/Ab A A#/Bb B/Cb C/B#

Любые две смежные ноты из списка образуют полутон. Любые две ноты, имеющие точно одну разделяющую их ноту, образуют тон. Октава состоит из 8 нот. Она начинается с любой ноты, и ноты, образующие октаву, составляют прогрессию тон-тон-полутон-тон-тон-тон-полутон. Например, октавы, начинающиеся соответственно с C и Db, составлены из следующих нот:
C D E F G A B C
Db Eb F Gb Ab Bb C Db

Кроме того, на октавы накладываются следующие ограничения:

1. Октава содержит каждую букву от A до G ровно один раз за исключением первой буквы октавы, которая совпадает с последней буквой октавы.

2. В октаве не могут находиться одновременно диез- и бемоль-ноты.

Нота, с которой начинается октава, называется тональностью октавы. Так, октавы, приведённые выше, — это октавы с тональностями C и Db соответственно. Транспозиция ноты из одной тональности в другую состоит в замене ноты из октавы в первой тональности на ноту в соответствующей позиции октавы другой тональности. Например, нота F в тональности C транспонируется в ноту Gb в тональности Db.

Формат входного файла

Каждая нота записывается на отдельной строке без пробелов в начале и конце строки. Пустые строки не допускаются. В первой строке записывается тональность первой октавы, во второй — тональность второй октавы. Далее перечисляются ноты, составляющие партитуру, которые нужно транспонировать.

Формат выходного файла

Если тональность первой октавы задана неправильно, то есть октавы с такой тональностью не существует, Ваша программа должна напечатать единственную строку `BAD` (пробелы в начале и конце строки не допускаются). Если тональность второй октавы задана неправильно, программа должна напечатать `WORSE`. Если обе тональности заданы неправильно, напечатайте `WORST`. Если оба ключа заданы правильно, программа должна напечатать транспонированные ноты по одной на строку, причём пробелы в начале и конце строки не допускаются. Если какая-либо из исходных нот не входит в октаву, на соответствующей строке должно быть напечатано `SKIP`.

Пример

bill.in	bill.out
C Db F	Gb
C B# A B	WORSE

Задача E. Сборы

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

Одна молодая и перспективная, но уже опытная команда повторно вышла в финал чемпионата мира по программированию. Так как результаты предыдущих официальных соревнований не устроили её тренеров, то, во-первых, было принято решение о проведении с утра 1 января K-дневных сборов, каждый день которых состоит из утренней и вечерней тренировок, а во-вторых, было решено проэкспериментировать с изначальной рассадкой участников. До начала серии тренировок изначальная рассадка команды перед компьютером была следующей: слева — гениальный математик G, в центре, за компьютером — капитан команды C, справа — великий тестер V.

В начале каждой утренней тренировки участник, на предыдущей тренировке (или перед серией, если тренировка первая) сидевший справа, меняется местами с участником, сидевщим в центре. В начале каждой вечерней тренировки участник, на предыдущей тренировке сидевший слева, меняется местами с участником, сидевшим в центре. Требуется определить рассадку участников после вечерней тренировки K-го дня сборов.

Формат входного файла

В первой строке записано целое число M ($1 \le M \le 12$) — количество тестовых примеров. Каждая из следующих M строк содержит целое число K ($1 \le K \le 1000$) — продолжительность сборов в днях.

Формат выходного файла

Для кажжого тестового примера выведите рассадку участников после вечерней тренировки последнего дня сборов — три заглавных латинских буквы G, C, V, записаных в порядке рассадки слева направо.

Пример

camp.in	camp.out
2 1 5	VGC CVG

Задача F. Шпионаж

Имя входного файла:	`spy.in`
Имя выходного файла:	`spy.out`
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

В распоряжение резидента разведки в Нагонии полковника Моргулиса попали совершенно секретные документы из Лаборатории по Изучению Поведения Атмосферы, из которых, в частности, следует, что слухи о так называемых НЛО начали планомерно запускаться в 1918 году по личному распоряжению У. Черчилля.

К сожалению, документы с важными расчётами попали резиденту в слегка искажённом виде. Во многих расчётах пропущена одна цифра или знак операции, более того, неизвестно, в каком именно месте произошло искажение. Например, выражение `1+23=24` могло превратиться в `1+2=24`, `123=24`, `1+2324` или даже `1+23=4`.

Согласно агентурным данным в оригинальном документе используются только целые числа, и операции проводятся только над целыми числами, причём ни в какой момент при вычислении выражения абсолютное значение промежуточного результата не может превышать 2^{15}. Целые числа записываются в десятичной системе счисления, перед отрицательным числом ставится знак «минус». Положительное число никогда не записывается со знаком «плюс». Число никогда не содержит незначащих нулей. Ноль всегда записывается как `0`. В выражении используется пять операций, обозначаемых `+`, `-`, `*`, `/` (деление нацело), `%` (остаток от деления). Операции умножения, деления и взятия остатка имеют более высокий приоритет, чем сложение и вычитание. Группа операций с равным приоритетом выполняется слева направо. Для изменения порядка выполнения операций могут использоваться скобки `(` и `)`. В выражении всегда присутствует знак равенства, после которого находится число. Операции деления и взятия остатка выполняются только над неотрицательными аргументами. Делитель никогда не равен нулю.

Напишите программу, которая вставит недостающий символ в выражение, если известно, что исходное выражение удовлетворяет всем условиям, описанным выше.

Формат входного файла

Во входном файле находится выражение, записанное на одной строке, причём пробелы в начале и конце строки не допускаются. Максимальная длина выражения — 250 символов. Пробелы внутри выражения не допускаются.

Формат выходного файла

Если ни при какой вставке пропущенного символа правильного выражения не получается, Ваша программа должна напечатать `NO`. Если существует более одного варианта вставки, приводящего к правильному выражению, программа должна напечатать `MANY`. Если существует единственный вариант вставки, программа должна напечатать получающееся выражение.

Пример

spy.in	spy.out
`2(-5+9)-3=5`	`2*(-5+9)-3=5`
`5+5=0`	`MANY`