Задача A. Угадай строку

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	150

Это — интерактивная задача.

Загадана непустая строка S, состоящая не более, чем из 10 строчных букв латинского алфавита. Ваша задача — определить строку S, делая запросы в виде маски, состоящей из строчных букв латинского алфавита или символов «*». Символ «*» обозначает любую (возможно, пустую) последовательность строчных букв латинского алфавита.

Взаимодействие с программой жюри

На каждый запрос участника программа жюри сообщает, соответствует ли эта маска загаданной строке S. Если участник получил положительный ответ на строку без символов «*», то считается, что он угадал строку S.

Гарантируется, что предоставленного количества запросов достаточно, чтобы определить загаданную строку.

Формат выходных данных

Каждый запрос должен содержать непустую строку длины не более 21, состоящую из строчных латинских букв или символов «*» (ASCII 42).

Количество запросов не должно превышать 150.

Не забывайте использовать команду `flush` после вывода каждой строки.

Формат входных данных

Каждый ответ на запрос выводится на отдельной строке. Строка с ответом содержит «`Yes`», если маска соответствует строке, «`No`», если не соответствует, и «`Exit`», если необходимо осуществить выход из программы.

Пример

стандартный поток вывода	стандартный поток ввода
aca	Yes
*a*a*a*a*	Yes
aca*	No
*aca	No
aba	No
aba*aba	Yes
abacaba	Exit

Задача B. Минное поле

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	10 000

Это — интерактивная задача.

Вы находитесь на прямоугольном клетчатом поле размера $N \times M$ ($N = 3, M = 3$). Поле является тороидальным. Это означает, что можно мысленно соедининить левую и правую границы поля, и, аналогично, верхнюю и нижнюю границы поля.

Некоторые клетки поля содержат мины, а одна из пустых клеток содержит клад. На мины наступать нельзя. Ваша цель — найти клад. Гарантируется, что начальная клетка не содержит ни мину, ни клад. За ход можно переместиться в смежную по стороне клетку, сделать циклический сдвиг строки или сделать циклический сдвиг столбца. Сдвигать можно только строку или столбец той клетки, в который Вы сейчас находитесь. При сдвиге смещается все: Ваше расположение, мины и клад.

Считается, что вы нашли клад, только если последовательность ваших ходов такова, что для любого начального положения клада вы в какой-то момент оказались с ним на одной позиции. Гарантируется, что найти клад всегда возможно.

Взаимодействие с программой жюри

В начале и после каждого хода программа жюри сообщает суммарное количество мин в строке и столбце, где находится участник, или просит выполнить выход из программы (если игрок уже нашел клад).

Формат выходных данных

Каждая строка должна содержать один символ «`l`», «`u`», «`r`» или «`d`», означающий движение влево, вверх, вправо и вниз соответственно или «`L`», «`U`», «`R`» или «`D`», означающих циклический сдвиг влево, вверх, вправо и вниз соответственно.

Общее количество символов, обозначающих движение или сдвиг, не должно превышать 10 000.

Не забывайте использовать команду `flush` после вывода каждой строки.

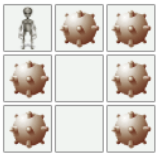
Формат входных данных

Каждая строка содержит одно число — общее количество мин или -1, если необходимо выполнить выход из программы.

Пример

стандартный поток вывода	стандартный поток ввода
	4
D	3
D	3
r	2
D	3
u	-1

Примеру соответствует поле:



Задача C. Сопротивление

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	12 секунд (15 секунд для Java)
Ограничение по памяти:	256 мегабайт

Это — интерактивная задача.

В задаче рассмотрим упрощенную версию популярной игры «Сопротивление». В игре принимают участие N игроков. В начале игры всем в случайном порядке раздаются карточки с ролями. Роли остаются неизвестными другим игрокам до конца игры. Есть две роли: активисты Сопротивления и шпионы Империи. Цель каждой стороны — получить три очка. Очки начисляются за выполнение или саботаж миссий.

Игра состоит из 3–5 миссий. В этой задаче только Вы решаете, кто пойдет на миссию. Больше никаких решений Вы не принимаете. Каждый игрок, идущий на миссию, решает, выполнить ее или саботировать (кто что выбрал — неизвестно). Активисты Сопротивления всегда выполняют миссию. Если все выполняют миссию, то очко получают активисты Сопротивления, иначе — шпионы Империи.

Количество шпионов в зависимости от количества участников:

Количество игроков:	5	6	7	8
Количество активистов Сопротивления:	3	4	4	5
Количество шпионов Империи:	2	2	3	3

Количество игроков, необходимых для миссии:

Количество игроков:	5	6	7	8
Миссия 1:	2	2	2	3
Миссия 2:	3	3	3	4
Миссия 3:	2	4	3	4*
Миссия 4:	3	3	4*	5*
Миссия 5:	3	4	4	5

(*) чтобы шпионы империи получили очко, необходимо как минимум два саботажа.

Миссии выдаются в том порядке, в котором они приведены в таблице выше.

Стратегия шпионов: каждый независимо от других выбирает для себя в начале игры случайно и равновероятно число от 0.7 до 1 — с такой вероятностью он будет саботировать миссию. Если у шпионов уже есть два очка, каждый из них обязательно саботирует.

Взаимодействие с программой жюри

Каждый тест состоит из 4 000 игр. В начале каждой игры программа жюри выводит количество игроков N. Далее следует описание раундов. В начале каждого раунда программа жюри просит назвать номера игроков, идущих на миссию. Программа участника должна выдать эти номера, а в ответ получить количество саботажей или признак конца игры.

Решения участников для этой задачи будут проверяться на одном тесте:

1. Во всех играх $n = 5$.

Решение участника считается правильным, если активисты Сопротивления выиграли не менее 49% игр.

Формат входных данных

Перед началом первого раунда игры «Сопротивление» программа жюри выводит целое положительное число N — количество игроков.

Формат раунда:

В начале раунда программа жюри сообщает участнику единственное положительное число K — число игроков, которые должны пойти на миссию.

В ответ на список участников миссии программе участника сообщается единственная строка, содержащая количество саботажей, или текст «WIN» или «LOSE», если активисты Сопротивления выиграли или проиграли соответственно.

Последняя строка содержит единственное число $N = -1$. Эту игру не надо обрабатывать, необходимо просто выйти из программы.

Формат выходных данных

Для каждого запроса списка участников миссии программа участника должна выдавать список номеров игроков A_1, A_2, …, A_K ($1 \leqslant A_i \leqslant N$, $A_i \neq A_j$ при $i \neq j$), которые пойдут на миссию.

Не забывайте использовать команду `flush` после вывода каждой строки.

Пример

стандартный поток ввода	стандартный поток вывода
5 2 1 3 1 2 LOSE 5 2 1 3 0 2 1 3 0 3 WIN -1	 3 5 1 2 3 1 4 1 2 1 4 5 4 5 1 3 5 1 3 5

Пример, указанный в условии, не соответствует первому тесту. В тестовом наборе первый тест состоит из 4 000 игр.

В первой игре шпионами были игроки с номерами 3 и 4, а во второй — 2 и 4.

Задача D. Робот

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт

Это — интерактивная задача.

Дано квадратное поле размера 81×81. Координаты поля — целые числа от 0 до 80. В центре поля (в клетке с координатам $(40, 40)$) стоит робот. Робот и игрок делают ходы по очереди: сначала робот двигается по полю в одном из четырех направлений, а следом игрок сжигает одну из клеток на поле. Цель робота — покинуть поле, выйдя за его пределы. Цель игрока — помешать роботу достичь своей цели.

Робот может переместиться только на одну клетку (расстояние, равное единице). Он не может наступать на сожженные клетки.

Игроку запрещается сжигать одну и ту же клетку несколько раз, а также сжигать клетку, где в данный момент находится робот.

Взаимодействие с программой жюри

Программа жюри и программа игрока ходят по очереди. Так как первым ходит робот, программа жюри выполняет ход первой. Она выводит либо «`E`», что означает, что робот сдается и игрок победил, либо направление и длину хода робота, после чего ждет ответного хода игрока. Программа игрока должна выводить координаты клетки, которую собирается сжигать.

Формат входных данных

Программа жюри может выводить команды «`L 1`», «`R 1`», «`U 1`», «`D 1`», «`E`». Если программа жюри вывела «`E`», то робот сдается и необходимо выполнить выход из программы. В противном случае команда обозначает перемещение робота: направление (влево, вправо, вверх или вниз), и в этом случае программа жюри ждет ответного хода игрока. Движение влево соответствует уменьшению номера столбца, вправо — увеличению, вверх — уменьшению номера строки, вниз — увеличению.

Формат выходных данных

После каждого ответа программы жюри, сообщающего о перемещении робота, программа игрока должна выдавать два целых числа от 0 до 80 — номер строки и номер столбца клетки на поле, которую игрок собирается сжечь.

Не забывайте использовать команду `flush` после вывода каждой строки.

Пример

стандартный поток ввода	стандартный поток вывода

`L 1`
`U 1`
`U 1`
`U 1`
`D 1`
`D 1`
`D 1`
`E`
	`40 38`
	`36 39`
	`37 38`
	`42 42`
	`41 39`
	`40 40`
	`39 39`

Задача E. Лифты

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	10 000

Это — интерактивная задача.

В здании N ($2 \leqslant N \leqslant 20$) этажей и K ($1 \leqslant K \leqslant 3$) лифтов. Каждый лифт можно вызвать на любом этаже, однако в лифте есть всего две кнопки: вверх и вниз. После нажатия некоторой кнопки i-й лифт перемещается ровно на A_i ($1 \leqslant A_i < N$) этажей в заданном направлении. Если невозможно переместиться на A_i шагов в заданном направлении, то нажатие кнопки игнорируется. Заранее числа A_i неизвестны.

Вам назначили встречу на некотором этаже, но Вы не знаете, на каком именно. Вы находитесь на первом этаже и должны попасть на эту встречу, используя только эти лифты. Для этого Вам требуется найти последовательность нажатий кнопок такую, что вне зависимости от значений A_i Вы посетите все этажи. Гарантируется, что можно добраться на любой этаж.

Взаимодействие с программой жюри

В самом начале программа жюри выводит количество этажей N и количество лифтов K. Далее в ответ на команду «вызвать лифт и поехать» программа жюри сообщает, сдвинулся ли лифт, или просит выйти из программы, если лифт посетил все этажи.

Формат выходных данных

Каждая строка должна содержать одно положительное или отрицательное число x ($1 \leqslant |x| \leqslant K$). Если число положительное, то это означает «вызвать лифт x и поехать вверх», иначе — «вызвать лифт $-x$ и поехать вниз».

Количество выводимых чисел не должно превышать 10 000.

Не забывайте использовать команду `flush` после вывода каждой строки.

Формат входных данных

Первая строка содержит два целых числа N и K. Следующие строки содержат «`Yes`», если лифт поехал, «`No`», если остался на месте, и «`Quit`», если необходимо выполнить выход из программы.

Пример

стандартный поток вывода	стандартный поток ввода
	6 2
1	Yes
1	Yes
1	No
-2	Yes
-2	No
-1	No
1	Yes
1	Quit

В этом примере $N = 6$, $K = 2$, $A_1 = 2$, $A_2 = 3$.

Задача F. Лабиринт

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	100 000

Это — интерактивная задача.

Лабиринт представляет собой прямоугольную сетку. Длина и ширина лабиринта не превышают 10. Любая клетка лабиринта либо свободна, либо содержит стену, либо содержит телепорт. Лабиринт содержит не более одной пары двусторонних телепортов. Каждый из телепортов принадлежит ровно одной паре.

Вы находитесь в лабиринте и у Вас есть один камень. Других камней в лабиринте нет. За один ход Вы можете попытаться перейти на клетку вверх, вниз, влево или вправо, попытаться оставить камень в клетке, в которой находитесь, или попытаться взять камень. Если игрок шагнет на клетку с телепортом из другой клетки, то он будет перемещен в клетку с парным ему телепортом и останется там до следующего шага. Если игрок стоит на клетке с телепортом и пытается сходить в стену, скинуть камень или поднять камень, он (игрок) остается на месте, а не перемещается по телепорту. Если камень положить в клетку с телепортом, то он не телепортируется. Ваша задача — определить количество клеток, не содержащих стены, до которых можно добраться.

Гарантируется, что невозможно выйти за пределы лабиринта. Гарантируется, что не существует двух соседних клеток, обе из которых содержат телепорт. Гарантируется, что рядом с любой клеткой с телепортом есть хотя бы одна пустая клетка. Гарантируется, что начальная клетка не содержит телепорт.

Взаимодействие с программой жюри

В ответ на любую команду участника, кроме команды вывода ответа, программа жюри либо сообщает, что он уперся в стену (в таком случае он останется на месте), либо сообщает тип клетки, в которой находится участник: в пустой клетке или в клетке с камнем. Пустые клетки и клетки с телепортом неразличимы. Если участник нашел ответ, то он может вывести его специальной командой и сразу выполнить выход из программы.

Формат выходных данных

Каждая строка должна содержать одну команду:

- `MOVE dir`: движение в направлении `dir`, где `dir` — это `UP` (вверх), `DOWN` (вниз), `LEFT` (влево), `RIGHT` (вправо).
- `DROP`: попытаться положить камень.
- `TAKE`: попытаться подобрать камень.
- `DONE x`: вывести ответ, где `x` — количество клеток, не содержащих стены, до которых можно добраться (включая исходную позицию).

Если в момент выполнения команды `DROP` камня у игрока не оказалось, или же в момент выполнения команды `TAKE` камня не оказалось в клетке, перемещение камня не происходит, а игроку просто сообщается тип клетки.

Общее количество всех команд не должно превышать 100 000.

Не забывайте использовать команду `flush` после вывода каждой строки.

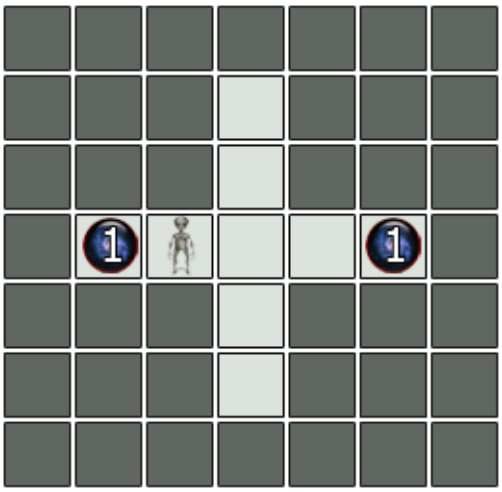
Формат входных данных

Каждая строка содержит `WALL`, если участник уперся в стену, `EMPTY CELL` — если он находится в пустой клетке без камня, `CELL WITH STONE` — если он находится в клетке с камнем.

Пример

стандартный поток вывода	стандартный поток ввода
MOVE RIGHT	EMPTY CELL
DROP	CELL WITH STONE
MOVE RIGHT	EMPTY CELL
MOVE RIGHT	EMPTY CELL
MOVE RIGHT	EMPTY CELL
MOVE RIGHT	CELL WITH STONE
MOVE UP	EMPTY CELL
MOVE UP	EMPTY CELL
MOVE UP	WALL
MOVE DOWN	EMPTY CELL
MOVE DOWN	CELL WITH STONE
TAKE	EMPTY CELL
MOVE DOWN	EMPTY CELL
MOVE DOWN	EMPTY CELL
MOVE DOWN	WALL
DONE 9	

Примеру соответствует следующий лабиринт:



Задача G. Отражения

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	2 000

Это — интерактивная задача.

Дана плоскость, на которой расстояние между точками (x_1, y_1) и (x_2, y_2) определяется следующим образом:

$$\sqrt{(T_1 \cdot x_1 - T_1 \cdot x_2)^2 + (T_2 \cdot y_1 - T_2 \cdot y_2)^2},$$

где T_1 и T_2 — неизвестные целые числа ($0 \leqslant |T_1|, |T_2| \leqslant 100$).

Вы управляете пером, которое рисует ломаную. Первоначально оно находится в начале координат. Также на плоскости есть 4 прямых. Вы можете зеркально отразить перо относительно любой прямой или провести следующее звено и узнать его длину.

Ваша задача — найти расстояние между заданными точками a и b.

Взаимодействие с программой жюри

Сначала программа жюри выводит число 2, описание прямых и координаты точек a и b.

Далее после команды «отразить» она выводит координаты пера, после команды «провести звено» — длину звена, после вывода ответа — правильный ответ или нет.

Участник обязан выходить из программы только в случае правильного ответа (в задаче разрешается ошибаться, вывод ответа является обычным запросом). Ответ должен отличаться от правильного не более, чем на 10^{-3} по абсолютной или относительной погрешности.

Формат выходных данных

Каждая строка должна содержать одну команду:

- `Reflect` x — зеркально отразить перо относительно прямой x.
- `Request` — нарисовать звено и запросить его длину.
- `Answer` d — вывести ответ, где d — расстояние между точками a и b.

Количество выводимых строк не должно превышать 2 000.

Не забывайте использовать команду `flush` после вывода каждой строки.

Формат входных данных

Первая строка содержит одно целое число 2. Далее следует описание четырех прямых: для каждой прямой в первой строке записаны координаты ее нормали, а во второй — координаты точки, через которую она проходит.

Все числа целые. Все координаты нормалей не превышают 100 по абсолютной величине, координаты точек не превышают 1 000 по абсолютной величине. Никакие две прямые не параллельны. Ни одна прямая не проходит через начало координат.

В следующих двух строках содержится по 2 целых числа, не превышающих по абсолютной величине 10 000 — координаты точек a и b соответственно.

Далее каждая строка содержит ответы на запросы:

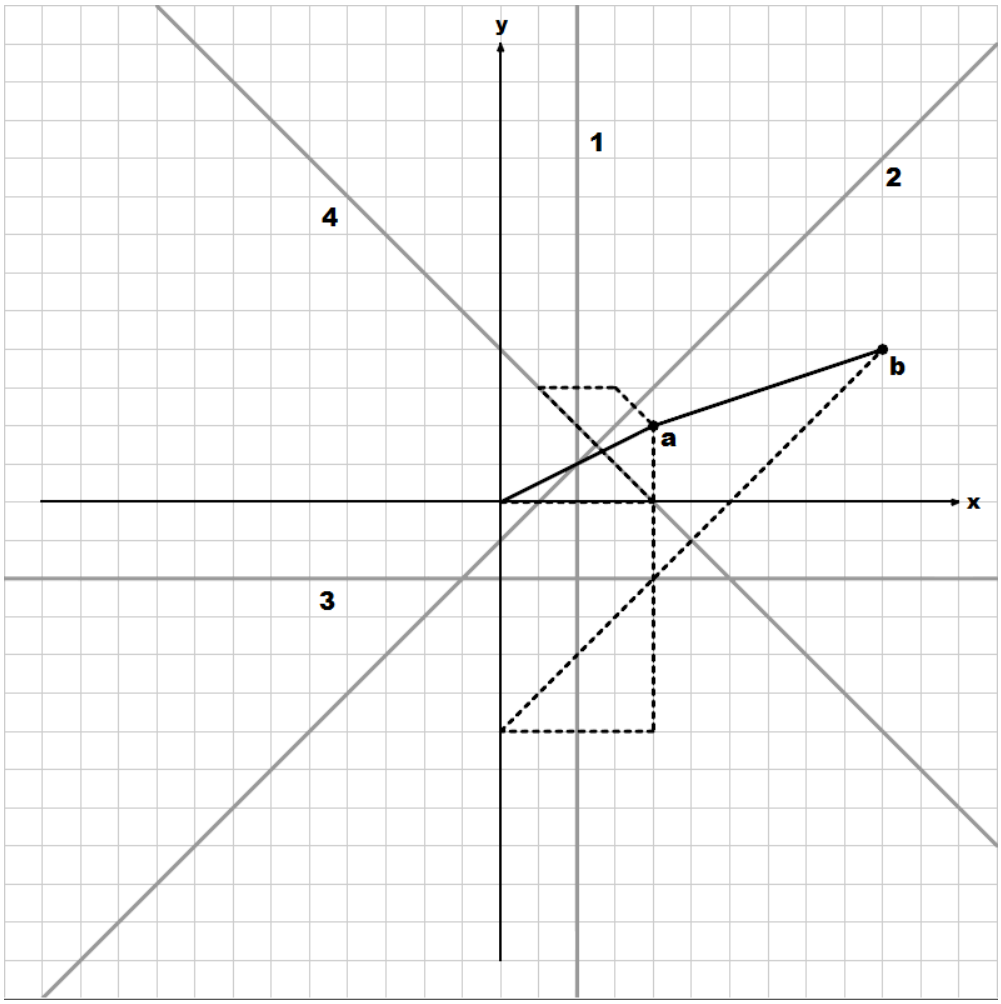
- `Reflect`: два вещественных числа с 9 знаками после десятичной точки — координаты пера.
- `Request`: одно вещественное число с 9 знаками после десятичной точки — длина звена.
- `Answer`: один символ «`N`», если ответ неправильный, или «`Y`», если правильный.

Пример

стандартный поток вывода	стандартный поток ввода
	2 1 0 2 0 -1 1 0 -1 0 1 0 -2 1 1 2 2 4 2 10 4
Answer 1	N
Answer 1.5	N
Reflect 1	4.000000000 0.000000000
Reflect 2	1.000000000 3.000000000
Reflect 1	3.000000000 3.000000000
Reflect 2	4.000000000 2.000000000
Request	8.944271910
Reflect 3	4.000000000 -6.000000000
Reflect 1	0.000000000 -6.000000000
Reflect 4	10.000000000 4.000000000
Request	12.649110641
Answer 12.64	Y

В приведенном примере $T_1 = 2$ и $T_2 = 2$.

Этому примеру соответствует следующее изображение:



Задача N. Стрельба

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт

Это — интерактивная задача.

Из некоторой точки на поверхности земли бросается шар радиусом 10 см со скоростью 100 м/с в некотором направлении под углом 45 градусов к поверхности земли. В начальный момент времени на расстоянии 1 050 метров от центра шара на поверхности земли стоит стрелок, желающий сбить шар. Так как расстояние очень большое, стрелок не видит, куда летит шар. Стрелок неподвижен.

В данной задаче можно считать землю плоской, стрелка — материальной точкой, а ускорение свободного падения равным 9.80665 м/с^2.

Из личных предпочтений и для удобства стрельбы стрелок мысленно рисует прямоугольную координатную мишень следующим образом. Мишень располагается между стрелком и шаром в плоскости, которая перпендикулярна прямой, соединяющей позицию стрелка и центр шара в начальный момент времени, и расположена на расстоянии 50 м от стрелка. Размер мишени — 1000 м в высоту и 2000 м в ширину. Она расположена непосредственно над поверхностью земли, и ее нижнее ребро симметрично относительно прямой, соединяющей стрелка и центр шара. На этой мишени вводится декартова система координат следующим образом: левый верхний угол (при взгляде от стрелка) имеет координаты $(-1000, 1000)$, а правый нижний — $(1000, 0)$.

Стреляет стрелок так: он выбирает некоторую точку на своей координатной мишени и стреляет в направлении этой точки. Выстрелом назовем луч, начинающийся в позиции стрелка и проходящий через выбранную им точку. После выстрела стрелок по рации узнает расстояние от выстрела до центра шара. В момент времени 0 шар лежит на земле, а стрелок находится на той же высоте, что и центр шара.

Первый выстрел производится через 0.5 секунды после броска. Каждый последующий выстрел производится через 0.5 секунды после предыдущего.

Итак, Вы — стрелок, и Ваша цель — попасть в шар, пока он не упал на землю.

Взаимодействие с программой жюри

После каждого выстрела программа жюри сообщает расстояние от луча до центра шара или просит выйти из программы, если участник попал в шар.

Формат выходных данных

Каждая строка должна содержать по два вещественных числа x и y ($-1000 \leqslant x \leqslant 1000$, $0 \leqslant y \leqslant 1000$), записанных в виде десятичной дроби — координаты точки на прямоугольнике.

Не забывайте использовать команду `flush` после вывода каждой строки.

Формат входных данных

Каждая строка содержит «`Miss` d», если стрелок промахнулся, где d — расстояние от луча до шара в метрах, или «`Quit`», если необходимо осуществить выход из программы.

Пример

стандартный поток вывода	стандартный поток ввода
0 0 0 1 0 3 0 6.7	Miss 34.130339059 Miss 46.215649475 Miss 38.336035338 Quit

В примере шар брошен к стрелку под углом, равным 45 градусам.

Задача I. Конвейер

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	10 000 команд `Next`, по 10 команд `Swap`

Это — интерактивная задача.

Перед Вами циклический конвейер, на котором лежат N ($20 \leqslant N \leqslant 100$) шаров. Некоторые из этих шаров — черные, а некоторые — белые. Вы можете видеть только пять подряд идущих слотов на конвейере. Вам разрешается менять местами любые 2 из видимых шаров в произвольном порядке или сдвигать весь конвейер против часовой стрелки на шаг $S = 1$. Вам необходимо определить количество черных шаров.

Первоначально никакие пять черных шаров не находятся рядом. Число N заранее не известно.

Взаимодействие с программой жюри

В начале и после каждой команды участника, кроме ответа, программа жюри выдает цвета пяти видимых шаров в порядке обхода по часовой стрелке. Если участник нашел ответ, то он может вывести его и сразу осуществить выход из программы.

Формат выходных данных

Каждая строка должна содержать «`Next`», если нужно привести конвейер в движение, «`Swap` a b», если нужно поменять местами видимые шары с номерами a и b (видимые шары нумеруются с 1 в порядке обхода по часовой стрелке), и «`Answer` x», если нужно вывести ответ, где x — количество черных шаров.

Количество операций вида `Next` не должно превышать 10 000. Количество подряд идущих операций вида «`Swap`» не должно превышать 10.

Не забывайте использовать команду `flush` после вывода каждой строки.

Формат входных данных

Каждая строка содержит по пять символов «`W`» или «`B`», обозначающих белый и черный шар соответственно. Шары выводятся в порядке обхода конвейера по часовой стрелке.

Пример

стандартный поток вывода	стандартный поток ввода
	BWWWW
Swap 1 3	WWBWW
Swap 3 5	WWWWB
Next	WWWBW
Next	WWBWB
Next	WBWBW
Swap 2 5	WWWBB
Next	WWBBW
Next	WBBWW
Next	BBWWW
Next	BWWWW
Next	WWWWB
Next	WWWBW
Next	WWBWW
Next	WBWWW
Swap 2 5	WWWWB
Next	WWWBW
Next	WWBWW
Next	WBWWB
Swap 2 3	WWBWB
Next	WBWBW
Next	BWBWW
Next	WBWWW
Next	BWWWW
Next	WWWWW
Answer 4	

В этом примере $N = 21$.
Начальное расположение шаров:

Задача J. Тест

Имя входного файла:	`стандартный поток ввода`
Имя выходного файла:	`стандартный поток вывода`
Ограничение по времени:	2 секунды (4 секунды для Java)
Ограничение по памяти:	256 мегабайт
Ограничение на количество запросов:	40

Это — интерактивная задача.

Вам необходимо правильно ответить на все вопросы теста, состоящего из 18 вопросов. В каждом вопросе по 4 варианта ответа: «a», «b», «c» и «d». Верные ответы любых двух подряд идущих вопросов различны.

У Вас есть несколько попыток правильно ответить на все вопросы теста. После каждой попытки Вы узнаете количество правильных ответов в этой попытке.

Взаимодействие с программой жюри

Для каждого набора ответов на вопросы теста программа жюри сообщает количество правильных ответов, либо сообщает о том, что необходимо осуществить выход из программы.

Формат выходных данных

Каждый запрос задается отдельной строкой S длины 18, состоящей из символов «a», «b», «c» и «d». i-й символ строки задает ответ на i-й вопрос.

Количество выводимых строк не должно превышать 40.

Не забывайте использовать команду `flush` после вывода каждой строки.

Формат входных данных

Каждая строка содержит одно число — количество правильных ответов или -1, если необходимо осуществить выход из программы.

Пример

стандартный поток вывода	стандартный поток ввода
aaaaaabbbbbbcccccc	5
bbbbbbddddddaaaaaa	7
abababababababbab	6
abacababababacaba	14
abacabaddddabacaba	16
abacabadadaabacaba	15
abacabadbbdabacaba	17
abacabadbcdabacaba	-1