Задача A. Крестики-нолики

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

В этой задаче предлагается предсказать результат игры «крестики-нолики» на доске 3×3 при оптимальной игре обеих сторон. Изначально доска пуста; первым ходом первый игрок ставит крестик в любую клетку, затем второй игрок ставит нолик в любую свободную клетку, и так далее. Игра заканчивается, когда на доске оказывается три крестика, стоящих в ряд (победа первого игрока), или три нолика, стоящих в ряд (победа второго игрока), или не остаётся пустых клеток (ничья). Делать ходы после конца игры нельзя.

Формат входного файла

Во входном файле даны три строки, каждая из которых содержит по три символа. Символ '`X`' (икс большое) означает крестик, символ '`O`' (о большое) — нолик, а '`.`' (точка) — пустую клетку поля.

Формат выходного файла

Если данная позиция получиться не могла, выведите '`Impossible`'.

Если в данной позиции крестики выигрывают при правильной игре или уже выиграли, выведите '`X`'.

Если в данной позиции нолики выигрывают при правильной игре или уже выиграли, выведите '`O`'.

Наконец, если получилась ничья, выведите '`Draw`'.

Примеры

xo.in	xo.out

`...`
`OX.`
`.X.`
	`X`

`OOO`
`...`
`XXX`
	`Impossible`

`XOX`
`.OX`
`.O.`
	`O`

`XOX`
`XXO`
`OXO`
	`Draw`

Задача B. Простой алгоритм

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

Даны два целых числа A и B в двоичной записи. С ними несколько раз производится следующее преобразование: вычисляются результаты операций XOR (побитовое исключающее или) $C = A \, xor \, B$ и AND (побитовое и) $D = A \, and \, B$, после чего A заменяется на C, а B — на $D \cdot 2$. Указанное преобразование повторяется до тех пор, пока одно из чисел A и B не станет равным нулю; при этом второе из этих чисел объявляется результатом работы алгоритма. Поскольку равенство нулю проверяется после преобразования, то последнее будет выполнено хотя бы один раз.

Зная A и B, найдите результат работы этого простого алгоритма.

Формат входного файла

В первой строке входного файла задано в двоичной записи число A, а во второй — число B. Каждое из этих чисел неотрицательно и не превосходит $2^{100\,000}$; числа не содержат в своей записи ведущих нулей.

Формат выходного файла

Выведите результат работы описанного алгоритма в двоичной записи без ведущих нулей.

Примеры

simple.in	simple.out
101 10	111
1100 1100	11000
1 0	1

Задача C. Обратный элемент

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

Дано целое число $n > 0$. Рассмотрим множество Z_n, элементами которого являются целые числа $0, 1, 2, \ldots, n-1$. Элемент q называется *обратным* к элементу p, если $(p \cdot q) \mod n = 1$.

Найдите обратный элемент q по заданным числам n и p.

Формат входного файла

В первой строке входного файла заданы два целых числа n и p через пробел ($1 \leqslant n \leqslant 10^9$, $0 \leqslant q < n$).

Формат выходного файла

Если у числа p нет в множестве Z_n обратного элемента, выведите -1.

В противном случае выведите q.

Примеры

inverse.in	inverse.out
3 2	2
5 2	3
8 4	-1
17 0	-1