

Задача 1. лягушка и кузнечик

Ограничение по времени: 0.5 секунд

В крайних клетках полосы шириной в одну клетку и длиной в N клеток сидят лягушка и кузнечик: лягушка в клетке № 1, кузнечик в клетке № N . Каждую секунду лягушка прыгает в сторону кузнечика, и одновременно кузнечик прыгает в сторону лягушки. Лягушка может прыгать только на две или на три клетки, кузнечик — только на одну или на две клетки. За какое наименьшее время они смогут оказаться в одной клетке?

Формат входных данных

Единственная строка входных данных содержит целое число N — длину клетчатой полосы ($2 \leq N \leq 2 \cdot 10^9$).

Формат выходных данных

Если лягушка и кузнечик могут оказаться в одной клетке, требуется вывести одно целое число — минимальное количество секунд, через которое они встретятся. Если они не смогут оказаться в одной клетке, требуется вывести число «-1» (без кавычек).

Система оценки

Решения, правильно работающие при $N \leq 30$, будут оцениваться в 30 баллов.

Решения, правильно работающие при $N \leq 10^5$, будут оцениваться в 50 баллов.

Примеры

стандартный ввод	стандартный вывод
5	1
9	2

Замечание

В первом примере лягушка может прыгнуть из клетки 1 в клетки 3 и 4, а кузнечик может прыгнуть из клетки 5 в клетки 3 и 4. Поэтому через 1 секунду они могут оказаться в одной клетке.

Во втором примере лягушка и кузнечик могут встретиться через 2 секунды. Например, лягушка прыгает в клетку 3, затем в клетку 6, а кузнечик прыгает в клетку 8, затем в клетку 6.

Задача 2. Центральные квадраты

Ограничение по времени: 1 секунда

Дан прямоугольник из $N \times M$ квадратов. Назовём квадраты на границе прямоугольника *крайними*. Расстоянием от какого-либо квадрата до края назовём количество перемещений, которое нужно сделать из данного квадрата в соседний по стороне квадрат, чтобы добраться от данного квадрата до крайнего квадрата. Квадраты с максимальным расстоянием до края, будем называть *центральными*. При этом квадрат может быть одновременно и крайним, и центральным.

На рисунке изображён прямоугольник для $N = 7$ и $M = 8$, в каждом квадрате которого записано расстояние от этого квадрата до края. У этого прямоугольника два центральных квадрата.

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	2	2	2	2	1	0
0	1	2	3	3	2	1	0
0	1	2	2	2	2	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

По данным N и M определите количество центральных квадратов в прямоугольнике.

Формат входных данных

Программа получает на вход два целых положительных числа записанных в разных строках, не превосходящих 10^9 — размеры прямоугольника.

Формат выходных данных

Программа должна вывести одно число — количество центральных клеток в данном прямоугольнике.

Система оценки

Решения, правильно работающие, когда входные числа не превосходят 100, будут оцениваться в 30 баллов.

Решения, правильно работающие, когда входные числа не превосходят 10^5 , будут оцениваться в 60 баллов.

Пример

стандартный ввод	стандартный вывод
7 8	2

Задача 3. Заказ в магазине

Ограничение по времени: 0.5 секунд

Решив заготовиться ручками на весь новый учебный год, Игорь подсчитал, что ему нужно M ручек.

В его любимом интернет-магазине есть удобная функция — он может сразу добавить в заказ упаковку из любого числа ручек от 1 до N . Правда, оказалось, что нельзя добавить в заказ две упаковки одного размера. Например, если Игорю нужно купить $M = 12$ ручек, а максимальное число ручек в упаковке $N = 10$, то Игорь может добавить в заказ упаковку из 7 ручек и упаковку из 5 ручек, но не сможет добавить две упаковки из 6 ручек.

Сформируйте заказ на M ручек, используя минимальное число различных упаковок.

Формат входных данных

Первая строка входных данных содержит число N — максимальный размер одной упаковки ($1 \leq N \leq 10^9$). Вторая строка входных данных содержит целое число M — необходимое количество ручек в заказе ($1 \leq M \leq 10^9$).

Формат выходных данных

Программа должна вывести одно или несколько чисел от 1 до N — размеры выбранных упаковок в любом порядке. Если имеется несколько возможных решений, то выведите любое из них. Если решения не существует, необходимо вывести одно число «0».

Система оценки

Решения, правильно работающие при $N \leq 10^5$, будут оцениваться в 40 баллов.

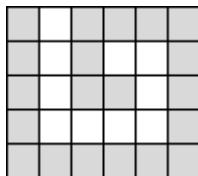
Примеры

стандартный ввод	стандартный вывод
10 12	5 7
2 5	0

Задача 4. Спираль

Ограничение по времени: 1 секунда

В левом верхнем углу прямоугольного поля размера $N \times M$ сидит Черепашка. Она хочет закрасить некоторые клетки по спирали, закручивающейся к центру, как на рисунке:



Определите, сколько клеток ей придётся закрасить.

Формат входных данных

Первая строка входных данных содержит число N — высоту прямоугольника, вторая строка содержит число M — ширину прямоугольника. Все числа — целые положительные и не превосходят 2×10^9 .

Формат выходных данных

Программа должна вывести одно целое число — количество клеток, покрашенных Черепашкой.

Обратите внимание, что ответ в этой задаче может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).

Система оценки

Решения, правильно работающие, когда числа N и M не превосходят 100, будут оцениваться в 40 баллов.

Решения, правильно работающие, когда числа N и M не превосходят 10^5 , будут оцениваться в 60 баллов.

Примеры

стандартный ввод	стандартный вывод
5 6	20
1 5	5

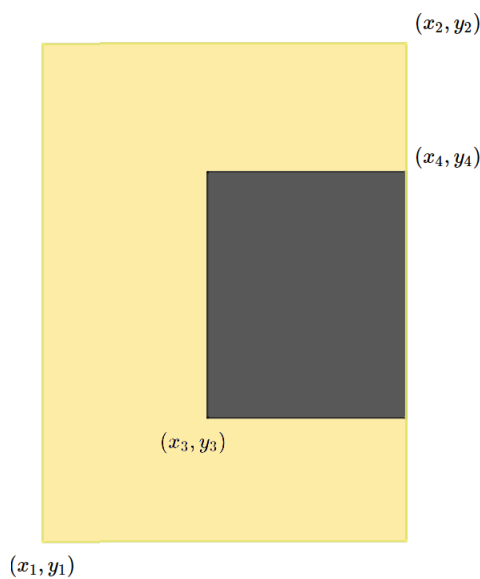
Задача 5. Надпись на табло

Ограничение по времени: 1 секунда

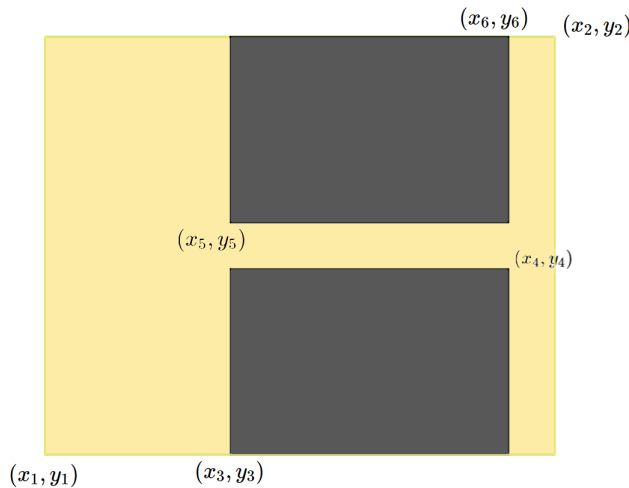
Вы получили доступ к одной из камер наблюдения в особо секретной организации. В зоне видимости камеры находится табло, с которого вы постоянно считываете информацию. Теперь вам нужно написать программу, которая по состоянию табло определяет, какая буква изображена на нём в данный момент. Табло представляет из себя квадратную таблицу, разбитую на $n \times n$ равных квадратных светодиодов. Каждый диод либо включён, либо выключен. Введём систему координат, направив ось OX вправо, а ось OY — вверх, приняв сторону диода равной 1.

На табло могут быть изображены только следующие буквы:

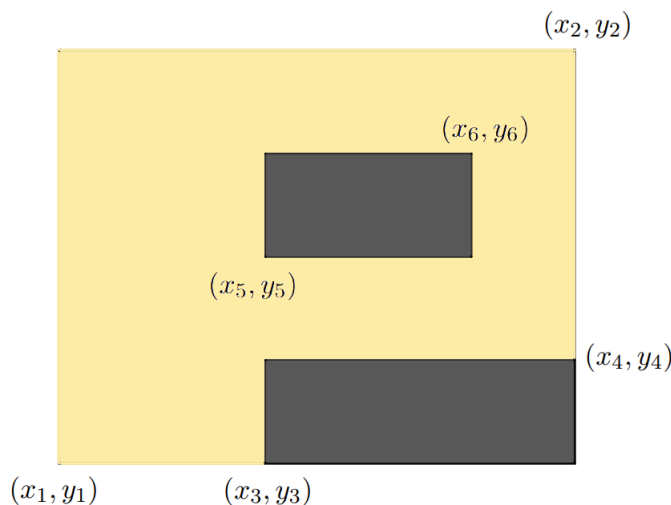
- **I** — прямоугольник из горящих диодов.
- **O** — прямоугольник из горящих диодов с углами (x_1, y_1) и (x_2, y_2) , внутри которого есть прямоугольник из выключенных диодов с координатами углов (x_3, y_3) и (x_4, y_4) . При этом границы выключенного прямоугольника не должны касаться внешнего, то есть $x_1 < x_3 < x_4 < x_2$ и $y_1 < y_3 < y_4 < y_2$.
- **C** — прямоугольник из горящих диодов с углами (x_1, y_1) и (x_2, y_2) , внутри которого есть прямоугольник из выключенных диодов с координатами углов (x_3, y_3) и (x_4, y_4) . При этом правая граница выключенного прямоугольника находится на правой границе внешнего прямоугольника, то есть $x_1 < x_3 < x_4 = x_2$ и $y_1 < y_3 < y_4 < y_2$.



- **L** — прямоугольник из горящих диодов с углами (x_1, y_1) и (x_2, y_2) , внутри которого есть прямоугольник из выключенных диодов с координатами углов (x_3, y_3) и (x_4, y_4) . При этом правые верхние углы выключенного прямоугольника и внешнего прямоугольника совпадают, то есть $x_1 < x_3 < x_4 = x_2$ и $y_1 < y_3 < y_4 = y_2$.
- **H** — прямоугольник из горящих диодов с углами (x_1, y_1) и (x_2, y_2) , внутри которого находятся 2 прямоугольника из выключенных диодов с координатами углов (x_3, y_3) , (x_4, y_4) у первого и (x_5, y_5) , (x_6, y_6) у второго. При этом выключенные прямоугольники должны иметь одинаковую ширину, находиться строго один под другим, один прямоугольник должен касаться верхней стороны, а другой прямоугольник должен касаться нижней стороны внешнего прямоугольника, то есть $x_1 < x_3 = x_5 < x_4 = x_6 < x_2$ и $y_1 = y_3 < y_4 < y_5 < y_6 = y_2$.



- **P** — прямоугольник из горящих диодов с углами (x_1, y_1) и (x_2, y_2) , внутри которого находятся 2 прямоугольника из выключенных диодов с координатами углов (x_3, y_3) , (x_4, y_4) у первого и (x_5, y_5) , (x_6, y_6) у второго. При этом правый нижний угол первого выключенного прямоугольника должен совпадать с правым нижним углом внешнего прямоугольника, а другой выключенный прямоугольник должен находиться строго выше и не касаться границ других прямоугольников, также левые границы двух выключенных прямоугольников должны совпадать, то есть $x_1 < x_3 = x_5 < x_6 < x_4 = x_2$ и $y_1 = y_3 < y_4 < y_5 < y_6 < y_2$.



- Любое другое состояние табло считается буквой **X**.

По виду табло определите, какая буква на нём изображена.

Формат входных данных

В первой строке входных данных находится одно число n ($1 \leq n \leq 10$) — сторона табло.

В следующих n строках находятся строки длины n из символов «.» и «#» — строки таблицы. «.» обозначает выключенный квадратный диод табло, а «#» — горящий.

Формат выходных данных

Программа должна вывести единственный символ: если данная таблица подходит под одно из описаний букв **I**, **O**, **C**, **L**, **H**, **P**, то выведите её (все буквы — английские). Если же данная таблица не подходит ни под какие условия, то выведите **X**.

Примеры

стандартный ввод	стандартный вывод
4 .##. .##. .##.	I
5 #...# .##. ..#.. .##. #...#	X