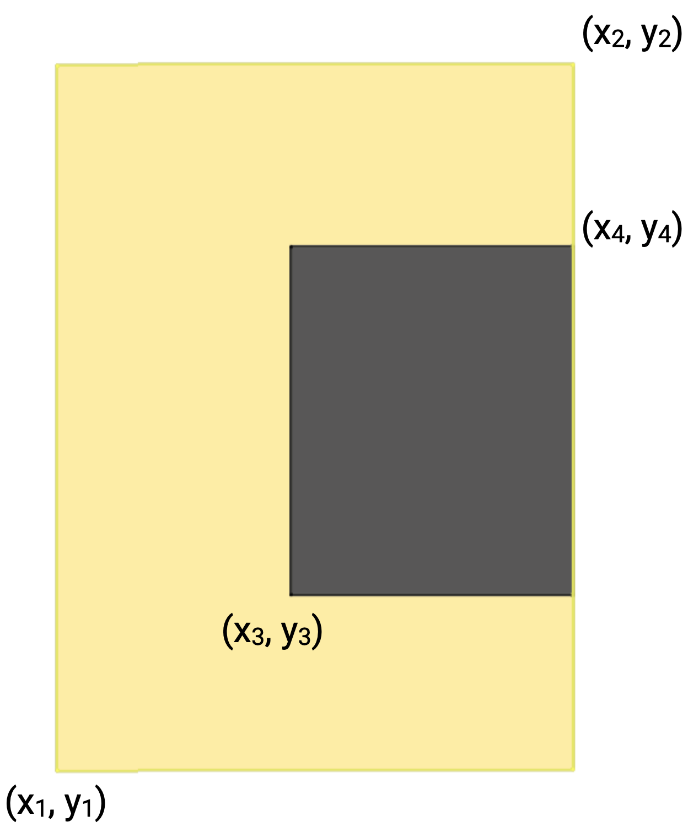
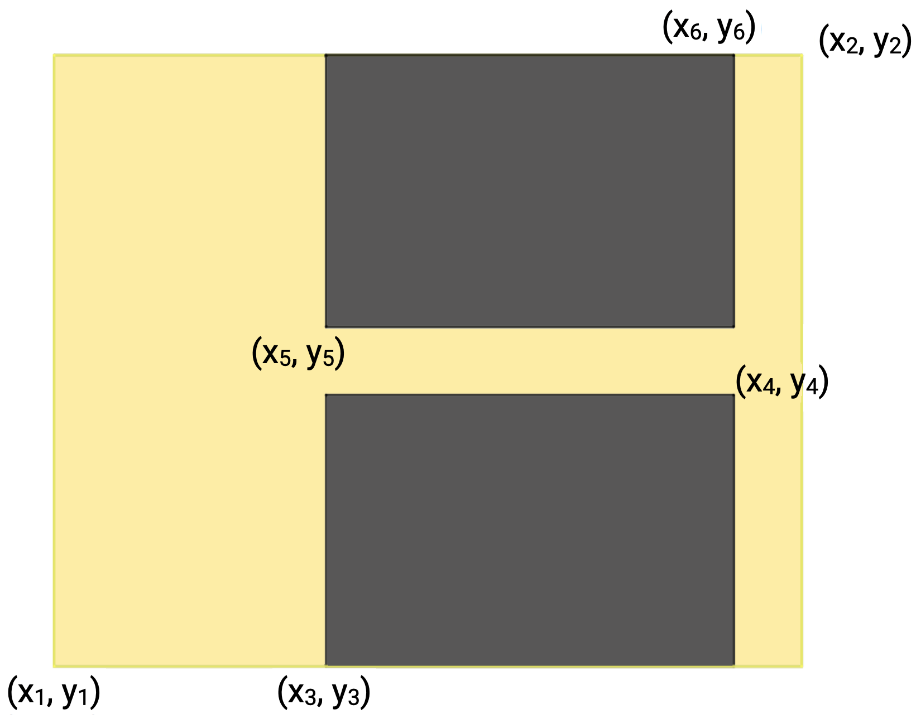
**№ 5**

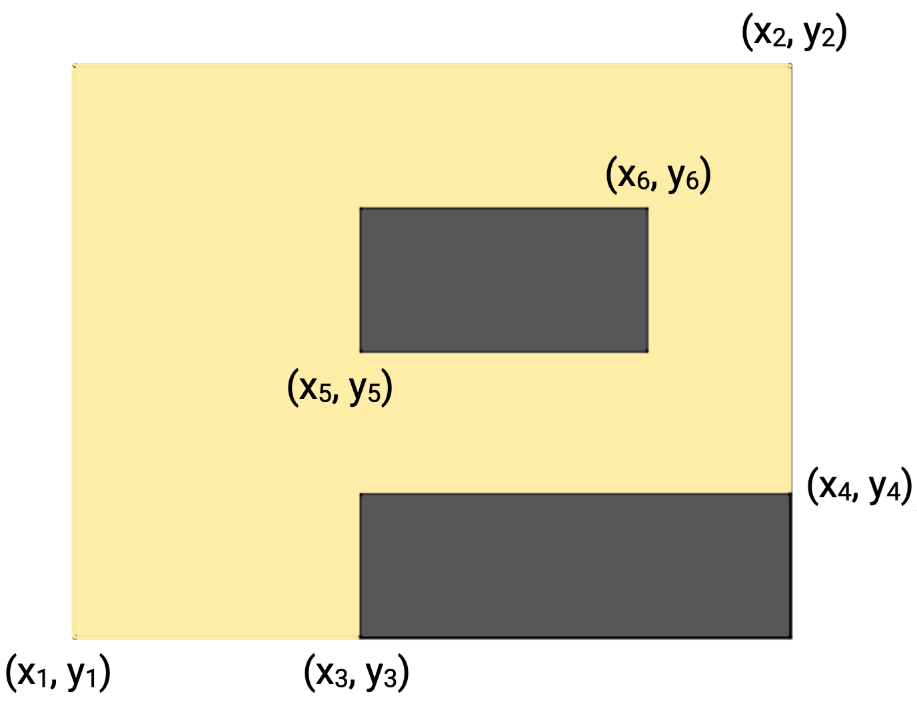
**Надпись на табло**

Ограничение по времени: 11 секунда

Вы получили доступ к одной из камер наблюдения в особо секретной огранизации. 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×nn×n равных квадратных светодиодов. Каждый диод либо включён, либо выключен. Введём систему координат, направив ось OXOX вправо, а ось OYOY —— вверх, приняв сторону диода равной 11.  
На табло могут быть изображены только следующие буквы:  
• II —— прямоугольник из горящих диодов.  
• OO —— прямоугольник из горящих диодов с углами (x1, y1)(x1, y1) и (x2, y2)(x2, y2), внутри которого есть прямоугольник из выключенных диодов с координатами углов (x3, y3)(x3, y3) и (x4, y4)(x4, y4). При этом границы внутреннего прямоугольника не должны касаться внешнего, то есть x1<x3<x4<x2x1<x3<x4<x2 и y1<y3<y4<y2y1<y3<y4<y2.  
• CC —— прямоугольник из горящих диодов с углами (x1, y1)(x1, y1) и (x2, y2)(x2, y2), внутри которого есть прямоугольник из выключенных диодов с координатами углов (x3, y3)(x3, y3) и (x4, y4)(x4, y4). При этом правая граница внутреннего прямоугольника находится на правой границе внешнего прямоугольника, то есть x1<x3<x4=x2x1<x3<x4=x2 и y1<y3<y4<y2y1<y3<y4<y2.  


• LL —— прямоугольник из горящих диодов с углами (x1, y1)(x1, y1) и (x2, y2)(x2, y2), внутри которого есть прямоугольник из выключенных диодов с координатами углов (x3, y3)(x3, y3) и (x4, y4)(x4, y4). При этом правые верхние углы внутреннего прямоугольника и внешнего прямоугольника совпадают, то есть x1<x3<x4=x2x1<x3<x4=x2 и y1<y3<y4=y2y1<y3<y4=y2.

• HH —— прямоугольник из горящих диодов с углами (x1, y1)(x1, y1) и (x2, y2)(x2, y2), внутри которого находятся 22 прямоугольника из выключенных диодов с координатами углов (x3, y3)(x3, y3) и (x4, y4)(x4, y4) у первого и (x5, y5),(x6, y6)(x5, y5),(x6, y6) у второго. При этом внутренние 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 x1<x3=x5<x4=x6<x2x1<x3=x5<x4=x6<x2 и y1=y3<y4<y5<y6=y2y1=y3<y4<y5<y6=y2.  


• PP —— прямоугольник из горящих диодов с углами (x1, y1)(x1, y1) и (x2, y2)(x2, y2), внутри которого находятся 22 прямоугольника из выключенных диодов с координатами углов (x3, y3)(x3, y3), (x4, y4)(x4, y4) у первого и (x5, y5),(x6, y6)(x5, y5),(x6, y6) у второго. При этом правый нижний угол первого внутреннего прямоугольника должен совпадать с правым нижним углом внешнего прямоугольника, а другой внутренний прямоугольник должен находиться строго выше и не касаться границ других прямоугольников. Также левые границы двух внутренних прямоугольников должны совпадать, то есть x1<x3=x5<x6<x4=x2x1<x3=x5<x6<x4=x2 и y1=y3<y4<y5<y6<y2y1=y3<y4<y5<y6<y2.  


• Любое другое состояние табло считается буквой XX.  
По виду табло определите, какая буква на нём изображена.

**Формат входных данных**

В первой строке входных данных находится одно число nn (1≤n≤10)(1≤n≤10) —— сторона табло.  
В следующих nn строках находятся строки длины nn 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,I, O, C, L, H, P, то выведите её (все буквы английские). Если же данная таблица не подходит ни под какие условия, то выведите XX.

**№ 1**

## Лягушка и кузнечик

Ограничение по времени: 0.50.5 секунды

В крайних клетках полоски шириной в одну клетку и длиной в NN клеток сидят лягушка и кузнечик: лягушка в клетке под номером 11, кузнечик в клетке под номером NN. 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 NN —— длину клетчатой полосы (2≤N≤2⋅109)(2≤N≤2⋅109).

### Формат выходных данных

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

### Система оценки

Решения, правильно работающие при N≤30N≤30, будут оцениваться в 3030 баллов.  
Решения, правильно работающие при N≤105N≤105, будут оцениваться в 5050 баллов.

### Пояснение

В первом примере лягушка может прыгнуть из клетки 11 в клетки 33 и 44, а кузнечик может прыгнуть из клетки 55 в клетки 33 и 44. Поэтому через 11 секунду они могут оказаться в одной клетке.  
Во втором примере лягушка и кузнечик могут встретиться через 22 секунды. Например, лягушка прыгает в клетку 33, затем в клетку 66, а кузнечик прыгает в клетку 88, затем в клетку 66.

##### Ввод

##### Вывод

5

1

9

2

**Код**

Python 3

keyboard\_arrow\_down



n = int(input())

ans = n//(3+2)

if n % 5 != 0:

ans +=1

print(ans)

Код из файла

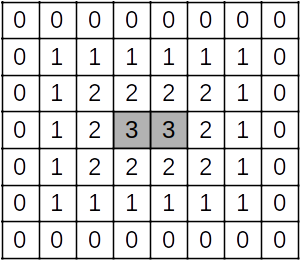
replayНачать сначала

Проверяем

Заменить ответ

## Центральные квадраты

Ограничение по времени: 11 секунда

Дан прямоугольник из N×MN×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=7N=7 и M=8M=8, в каждом квадрате которого записано расстояние от этого квадрата до края. У этого прямоугольника два центральных квадрата.  
  
По данным NN и MM определите количество центральных квадратов в прямоугольнике.

## Формат входных данных

Программа получает на вход два целых положительных числа, записанных в разных строках, не превосходящих 109109 —— 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 100100, будут оцениваться в 3030 баллов.  
Решения, правильно работающие, когда входные числа не превосходят 105105, будут оцениваться в 6060 баллов.

##### Ввод

##### Вывод

7

8

2

**Код**

Python 3

keyboard\_arrow\_down



n = int(input())

m = int(input())

if n in [0,1,2] or m in [0,1,2]:

print(0)

elif m>n:

if n%2==0:

print((m-n+2)\*2)

else:

print((m-n+1))

else:

if m%2==0:

print((n-m+2)\*2)

else:

print((n-m+1)\*1)

Код из файла

## Заказ в магазине

Ограничение по времени: 0.50.5 секунды

Решив запастись ручками на весь новый учебный год, Игорь подсчитал, что ему нужно MM ручек. В его любимом интернет‑магазине есть удобная функция —— он может сразу добавить в заказ упаковку из любого числа ручек от 11 до NN. Правда, оказалось, что нельзя добавить в заказ две упаковки одного размера. Например, если Игорю нужно купить M=12M=12 ручек, а максимальное число ручек в упаковке N=10N=10, то Игорь может добавить в заказ упаковку из 77 ручек и упаковку из 55 ручек, но не сможет добавить две упаковки из 66 ручек.  
Сформируйте заказ на MM ручек, используя минимальное число различных упаковок.

### Формат входных данных

Первая строка входных данных содержит число NN —— максимальный размер одной упаковки (1≤N≤109)(1≤N≤109).  
Вторая строка входных данных содержит целое число MM —— необходимое количество ручек в заказе (1≤M≤109)(1≤M≤109).

### Формат выходных данных

Программа должна вывести одно или несколько чисел от 11 до NN —— размеры выбранных упаковок в любом порядке. Есть имеется несколько возможных решений, то выведите любое из них. Если решения не существует, необходимо вывести одно число 00.

### Система оценки

Решения, правильно работающие при N≤30N≤30, будут оцениваться в 3030 баллов.  
Решения, правильно работающие при N≤105N≤105, будут оцениваться в 5050 баллов.

##### Ввод

##### Вывод

10

12

5

7

2

5

0

**Код**

Python 3

keyboard\_arrow\_down



n = int(input())

m = int(input())

ost = n+1

osts = []

while ost != 0:

if m > ost:

ost-=1

else:

ost = m

osts.append(ost)

break

m -= ost

osts.append(ost)

if osts[-1] == 0:

print(0)

else:

for i in osts:

print(i)

Код из файла