Bài 1:

Output: Tọa độ của 2 hình vuông ABCD và ABEF, mỗi hình vuông ghi trên 1 hàng, tọa độ của mỗi điểm có kèm ngoặc tròn, 4 điểm cách nhau bởi khoảng trắng, giữa 2 tọa độ của một điểm cách nhau bởi 1 dấu phẩy và 1 khoảng trắng.  
  
VÍ DỤ  
  
Input: 0 0 0 1  
  
OUTPUT: (0, 0) (0, 1) (1, 1) (1, 0)  
  
(0, 0) (0, 1) (-1, 1) (-1, 0)

xa,ya,xb,yb = map(float, input().split())

def lamtron(k):

if k == int(k):

n = int(k)

elif k == 0:

n = 0

else:

n = round(k,2)

return n

import math

vx = xa - xb

vy = ya - yb

tx = -vy

ty = vx

c = -tx\*xa - ty\*ya

d = round(math.sqrt((xa-xb)\*\*2+(ya-yb)\*\*2),4)

t = round(abs((d\*\*2-c-tx\*xa-ty\*ya)/(tx\*tx+ty\*ty)),4)

xd = round(xa + tx\*t,4)

yd = round(ya + ty\*t,4)

t1 = round(abs((d\*\*2-c-tx\*xb-ty\*yb)/(tx\*tx+ty\*ty)),4)

xc = round(xb + tx\*t,4)

yc = round(yb + ty\*t,4)

xc1 = (2\*xb - xc)

yc1 = (2\*yb - yc)

xd1 = (2\*xa - xd)

yd1 = (2\*ya - yd)

print('('+str((lamtron(xa)))+', '+str(lamtron(ya))+') '+'('+str(lamtron(xb))+', '+str(lamtron(yb))+') '+'('+str(lamtron(xc1))+', '+str(lamtron(yc1))+') '+'('+str(lamtron(xd1))+', '+str(lamtron(yd1))+')') print('('+str(lamtron(xa))+', '+str(lamtron(ya))+') '+'('+str(lamtron(xb))+', '+str(lamtron(yb))+') '+'('+str(lamtron(xc))+', '+str(lamtron(yc))+') '+'('+str(lamtron(xd))+', '+str(lamtron(yd))+')')

Bài 2:

Có 04 số nguyên dương phân biệt, trong đó tổng của 02 số bất kỳ sẽ chia hết cho a và tổng của 03 số bất kỳ sẽ chia hết cho b. Tổng nhỏ nhất của 04 số nguyên dương đó là bao nhiêu.  
  
Câu hỏi này thì thầy có câu trả lời nhưng vì tất cả các bạn trong lớp đều xứng đáng được vinh danh nên thầy sẽ dành cơ hội cho các bạn. Mỗi bạn tạo 5 test case với input và output như sau:  
  
INPUT  
Hai số nguyên dương a và b viêt trên cùng một hàng, cách nhau bởi khoảng trắng, giá trị của mỗi số không quá 200.  
  
OUTPUT  
Tổng nhỏ nhất của 04 số nguyên dương mà đề mô tả.

p = []

while True:

try:

line = input()

except EOFError:

break

for i in line.split():

if i != '':

p.append(int(i))

x = p[0]

y = p[1]

def ucln(a,b):

while a != b:

if a > b:

a -= b

else:

b-=a

return a

bcnn = int((x\*y)/ucln(x,y))

for i in range(1,1000):

a = i

b = a + bcnn

c = b + bcnn

d = c + bcnn

if(a+b)%x == 0 and (a+c)%x == 0 and (a+d)%x == 0 and (b+c)%x == 0 and (b+d)%x==0 and (c+d)%x==0 and (a+b+c)%y==0 and (a+b+d)%y==0 and(a+c+d)%y==0 and(b+c+d)%y==0:

print(a+b+c+d)

exit()

Bài 3:

MÔ TẢ BÀI TOÁN  
  
  
Khi bạn đốt cháy hết 1 cây nến thì sẽ còn sót lại phần sáp thừa. Có thể dùng makeNew phần sáp thừa này để tạo thành 1 cây nến mới. Rồi lại đến lượt cây nến mới này được đốt và để lại phần thừa khác.  
  
Giả sử, ban đầu bạn có candles cây nến. Hỏi tổng cộng có bao nhiêu cây nến mà bạn có thể đốt, biết rằng bạn có thể tạo ra những cây nến mới từ những phần sáp thừa.  
  
Ví dụ: Ban đầu bạn có candles = 5 cây nến, và cứ makeNew = 2 phần thừa sẽ tạo được 1 cây nến mới, khi đó output của bài toán là 9, nghĩa là bạn có thể đốt tổng cộng 9 cây nến. Cụ thể là:  
  
- Đốt 5 cây nến, có được 5 phần thừa. Tạo được 2 cây nến mới từ 4 phần thừa (còn dư 1 phần thừa )  
  
- Đốt 2 cây nến mới tạo được 2 phần thừa, tổng cộng ta có 3 phần thừa  
  
- Tạo ra 1 cây nến mới dùng 2 phần thừa ( còn dư 1 phần thừa)  
  
- Đốt cây nến mới tạo được 1 phần thừa khác (tổng cộng lúc này ta có 2 phần thừa)  
  
- Tạo được 1 cây nến từ 2 phần thừa còn lại. Đốt cháy cây nến cuối cùng  
  
- Tổng cộng: bạn có thể đốt: 5 +2+1+1 = 9 nến, đó là đáp án cuối cùng  
  
INPUT  
  
- Một số nguyên candles là số cây nến ban đầu bạn có với 1 ≤ candles ≤ 15  
  
- Một số nguyên makeNew chỉ số phần sáp thừa mà bạn có thể dùng để tạo ra một cây nến mới, 2 ≤ makeNew ≤ 5  
  
OUTPUT  
  
- Một số nguyên chỉ số cây nến có thể đốt

a = int(input())

b = int(input())

du = a

cay = a

while du > 0:

if du >= b:

cay+= int(du/b)

else:

break

du += int(du/b) - int(du/b)\*b

print(cay)