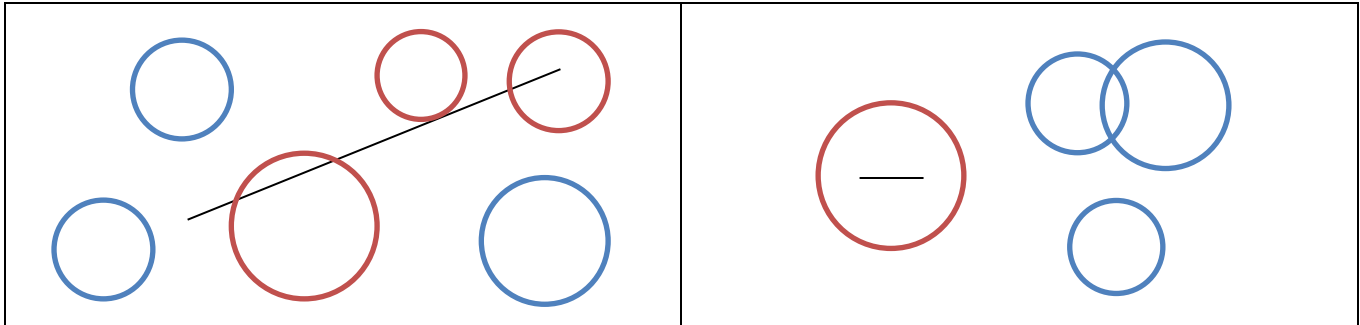


กำหนดให้มี “ส่วนของเส้นตรง” ที่มีจุดเริ่มต้นและจุดสิ้นสุดอยู่ในปริภูมิสองมิติ และวงกลม n วง ให้หาว่า มีวงกลมหมายเลขใดบ้างที่ “ส่วนของเส้นตรง” ลากผ่าน ซ้อนทับ หรือสัมผัสกับวงกลมนั้น จากภาพด้านล่าง วงกลมสีแดงคือคำตอบของปัญหาข้อนี้ **ให้นิสิตศึกษาการทำงาน** ของคลาสวงกลม Circle และคลาสส่วนของเส้นตรง Line จากนั้นให้เติมคำสั่งที่หายไป และเขียนส่วนหลักของโปรแกรมให้สมบูรณ์



```
import math

class Circle:

    def __init__(self, index, x, y, r):
        self.index = _____
        self.x = _____
        self.y = _____
        self.r = _____

    def line_intersection(self, line):
        # หาจุดตัดของ "เส้นตรง" และวงกลม ค่าที่ return เป็น tuple ของจำนวนจุดตัด ตามด้วยพิกัดของจุดตัดที่เป็นคำตอบ
        if line.x1 != line.x2:
            # line: y = Mx+B
            M = (line.y2-line.y1)/(line.x2-line.x1)
            B = line.y1-M*line.x1
            # แก้สมการจาก (x-self.x)**2+(y-self.y)**2 = self.r**2
            a = M**2+1
            b = -2*self.x + 2*M*(B-self.y)
            c = self.x**2+(B-self.y)**2-self.r**2
            if b**2 > 4*a*c:
                ans1 = (-b + math.sqrt(b**2-4*a*c))/2/a
                ans2 = (-b - math.sqrt(b**2-4*a*c))/2/a
                return (2, (ans1, M*ans1+B), (ans2, M*ans2+B))
            elif b**2 == 4*a*c:
                return (1, (-b/2/a, M*(-b/2/a)+B))
            else:
                return (0,)
        else:
            if self.r**2 == (line.x1-self.x)**2:
                return (1, (line.x1, self.y))
            elif self.r**2 > (line.x1-self.x)**2:
                rr = math.sqrt(self.r**2 - (line.x1-self.x)**2)
                ans1 = rr+self.y
                ans2 = rr-self.y
                return (2, (line.x1, ans1), (line.x1, ans2))
            else:
                return (0,)

    def contain_point(self, px, py):
        # ทดสอบว่าจุดอยู่ในวงกลมหรือไม่ ถ้าอยู่จะคืน True ถ้าไม่อยู่ จะคืน False
        if distance(self.x, self.y, px, py) <= self.r:
            return _____
        return _____
```

```

def line_in_circle(self,line):
    # คืน True ถ้า "ส่วนของเส้นตรง" ตัด ซ้อนทับ หรือสัมผัสวงกลม นอกจากนั้นให้คืน False
    if _____: # ถ้าจุดปลายของ "ส่วนของเส้นตรง" ซักข้างอยู่ในวงกลม ให้คืนค่า True
        return True
    ret = self.line_intersection(line) # หาคำตอบว่า "เส้นตรง" ตัดกับวงกลมกี่จุด
    if ret[0] == 0: # ผลเฉลยของสมการบอกว่า ไม่มีจุดตัดของวงกลมกับ "เส้นตรง" แสดงว่าไม่มีจุดตัดกับ "ส่วนของเส้นตรง" ด้วย
        return _____
    if ret[0] == 1: # ผลเฉลยของสมการบอกว่า "เส้นตรง" สัมผัสวงกลม 1 จุด
        # ให้เช็คด้วยว่าจุดนั้นอยู่ใน "ส่วนของเส้นตรง" หรือไม่ ถ้าอยู่ให้คืนค่า True ถ้าไม่อยู่ให้คืนค่า False
        return _____
    if ret[0] == 2: # ผลเฉลยของสมการบอกว่า "เส้นตรง" ตัดวงกลม 2 จุด
        # ให้เช็คด้วยว่า 2 จุดนั้น มีสักจุดอยู่ใน "ส่วนของเส้นตรง" หรือไม่ ถ้าอยู่ให้คืนค่า True ถ้าไม่อยู่ให้คืนค่า False
        return _____

def distance(x1,y1,x2,y2):
    # หาระยะห่างระหว่างจุดสองจุด
    return ((x1-x2)**2+(y1-y2)**2)**0.5

class Line:

    def __init__(self,x1,y1,x2,y2):
        self.x1 = _____
        self.y1 = _____
        self.x2 = _____
        self.y2 = _____

    def contain_point(self,px,py):
        # คืน True ถ้าจุดอยู่บน "ส่วนของเส้นตรง" คืน False ถ้าจุดไม่อยู่บน "ส่วนของเส้นตรง"
        if not (min(self.y1,self.y2) <= py <= max(self.y1,self.y2)):
            return False
        if self.x1 == self.x2:
            return px == self.x1
        else:
            return abs((self.y1-py)*(self.x2-self.x1)-(self.y2-self.y1)*(self.x1-px))<1e-5

# ส่วนของโปรแกรมหลัก

n = int(input().strip())
line_input = [int(e) for e in input().strip().split()]
line = _____

output = []

for i in range(n):
    circle_input = [int(e) for e in input().strip().split()]
    circle = _____
    if _____: # "ส่วนของเส้นตรง" ตัดหรือสัมผัสวงกลม
        output.append(_____) # เติมหมายเลขวงกลมลงใน output

if len(output) == 0:
    print('Not Found')
else:
    print(' '.join([str(e) for e in output]))

```

หมายเหตุ

- “เส้นตรง” ไม่เหมือนกับ “ส่วนของเส้นตรง” เพราะว่า “ส่วนของเส้นตรง” มีจุดเริ่มต้นและจุดสิ้นสุด คือปลายทั้งสองด้าน
- การทำงานของเมทอด line_intersection จะแก้สมการโดยหาจุดตัดระหว่างวงกลมกับ “เส้นตรง”
- เมทอด line_in_circle จะต้องตอบว่า “ส่วนของเส้นตรง” ทับหรือสัมผัสวงกลมหรือไม่ ดังนั้นการใช้คำตอบจากเมทอด line_intersection จะต้องตรวจสอบด้วยว่า จุดตัดที่ได้มา อยู่บน “ส่วนของเส้นตรง” หรือไม่
- เมทอด contain_point ในคลาส Line จะตรวจสอบว่า จุดอยู่บน “ส่วนของเส้นตรง” หรือไม่

ข้อมูลนำเข้า

บรรทัดแรกแสดงจำนวนวงกลม

บรรทัดที่สองแสดงจุดเริ่มต้นและจุดสิ้นสุดของส่วนของเส้นตรง โดยแสดงเป็นจำนวนเต็ม 4 ค่า คือ พิกัด x, y ของจุดเริ่มต้น และพิกัด x, y ของจุดสิ้นสุดของส่วนของเส้นตรง

n บรรทัดถัดไป แสดงข้อมูลของวงกลมแต่ละวง ซึ่งประกอบด้วยจำนวนเต็ม 4 จำนวน คำนด้วยเครื่องหมายเว้นวรรค แสดงหมายเลขประจำวงกลม พิกัดจุดศูนย์กลางวงกลมในแนวแกน x พิกัดจุดศูนย์กลางวงกลมในแนวแกน y และรัศมี

ข้อมูลส่งออก

แสดงหมายเลขของวงกลมที่ส่วนของเส้นตรงลากผ่านหรือแตะ โดยเรียงลำดับวงกลมที่รับเข้ามา

หากส่วนของเส้นตรงนั้นไม่ผ่านหรือสัมผัสวงกลมใดเลย ให้แสดงคำว่า **Not Found**

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
1 1 1 10 10 10 2 3 5	10
2 -10 -10 -12 -13 10 2 3 5 1 4 5 6	Not Found
2 2 3 4 5 10 2 3 1 1 4 5 1	10 1
3 -2 -2 -3 -3 10 2 3 4 1 4 5 3 2 -2 -3 4	2