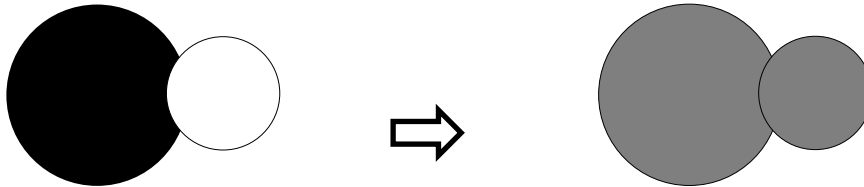


กำหนดให้มีวงกลมกระจายตัวอยู่ในปริภูมิสองมิติ วงกลมแต่ละวงจะมีปริมาณสีอยู่ข้างในซึ่งมีค่าตั้งแต่ 0 ถึง 1 โดยหากวงกลมมากกว่าหนึ่งวงแตะกันหรือสัมผัสกัน สีจะกระจายตามพื้นที่ของวงกลม เช่น วงกลมสองวงด้านล่าง วงกลมใหญ่สีดำ มีพื้นที่ $\pi(1.2)^2 = 4.524$ มีความสว่างของสี 0.0 ในขณะที่วงกลมวงเล็กสีขาว มีพื้นที่ $\pi(0.75)^2 = 1.767$ มีความสว่างของสีเท่ากับ 1.0 เมื่อรวมทั้งสองเข้าด้วยกัน จะกลายเป็นวงที่มีความสว่างเท่ากับ $\frac{(4.524)(0)+(1.767)(1)}{4.524+1.767} = 0.281$



หากมีวงกลมสามวงแตะหรือสัมผัสกัน โดยแต่ละวงมีพื้นที่ 2, 3, และ 4 มีความสว่างเท่ากับ 0, 0.5 และ 1.0 ตามลำดับ เมื่อกระจายสีแล้ว ทั้งสามวงจะมีความสว่างเท่ากับ $\frac{(2)(0)+(3)(0.5)+(4)(1.0)}{2+3+4} = 0.611$

ให้นักศึกษาเขียนโปรแกรมเพื่อรับข้อมูลวงกลม และจุดหนึ่งจุด จากนั้นให้หาว่า จุดนั้นควรมีความสว่างสีเท่ากับเท่าใด

นักศึกษาควรใช้คลาส Circle ด้านล่างให้เป็นประโยชน์

```
import math

class Circle:

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

    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 touch(self,other):
        # ทดสอบว่าวงกลมสองวง ซ้อนทับกัน หรือว่า แตะกัน หรือไม่
        if _____:
            return 'Overlap'
        elif _____:
            return 'Touch'
        return 'Free'

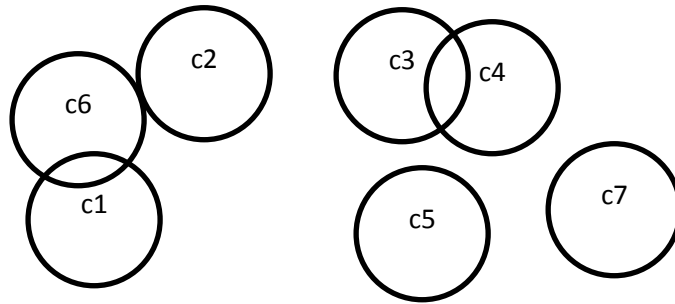
    def area(self):
        return _____ # คืนพื้นที่วงกลม

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

# ส่วนของโปรแกรมหลักให้เขียนต่อเอง
```

คำแนะนำ 1 (ถ้ารู้วิธีทำแล้ว จะไม่อ่านก็ได้)

ให้นิสิตสร้างลิสต์ซ้อนลิสต์ของวงกลม ที่แต่ละลิสต์ย่อยเก็บกลุ่มวงกลมที่เชื่อมกัน ตัวอย่างเช่น รูปนี้จะเก็บลิสต์ [[c1, c2, c6], [c3, c4], [c5], [c7]] (ลำดับของวงกลมไม่มีผล)



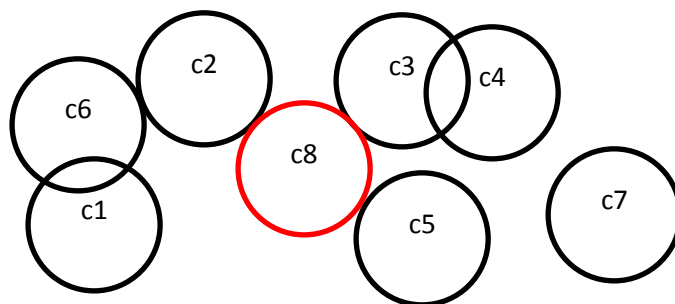
การสร้างลิสต์ซ้อนลิสต์นี้ จะทำโดยการเติมวงกลมเข้าไปทีละวง ให้เช็คทีละลิสต์ย่อยว่า วงกลมที่เพิ่มเข้ามาใหม่ เชื่อมกับวงกลมในลิสต์ย่อยนี้หรือไม่

- ถ้าไม่เชื่อมกับทุกลิสต์ย่อยเลย ให้สร้างลิสต์ย่อยใหม่ที่มีเฉพาะวงกลมวงนี้ แล้วนำไปต่อท้าย
- ถ้าเชื่อมกับบางลิสต์ย่อย ให้นำลิสต์ย่อยที่เชื่อมกับวงกลมใหม่ทั้งหมด มารวมกันทั้งหมดเป็นลิสต์ย่อยใหม่

ตัวอย่างด้านบน จะมีการสร้างลิสต์ซ้อนลิสต์ตามขั้นตอนดังนี้

- [[c1]]
- [[c1], [c2]]
- [[c1], [c2], [c3]]
- [[c1], [c2], [c3, c4]]
- [[c1], [c2], [c3, c4], [c5]]
- [[c1, c2, c6], [c3, c4], [c5]]
- [[c1, c2, c6], [c3, c4], [c5], [c7]]

หากมีเพิ่ม c8 ซึ่งจะเชื่อมกับกลุ่ม [c1, c2, c6], [c3, c4], [c5] จะทำให้ได้ผลลัพธ์สุดท้ายเป็น [[c1, c2, c3, c4, c5, c6, c8], [c7]]



เมื่อได้ลิสต์ซ้อนลิสต์นี้แล้ว ก็ตรวจสอบว่าจุดที่ให้มาอยู่ในกลุ่มของวงกลมใด และคำนวณค่าสื่อออกมา

(จริง ๆ แล้วมีวิธีการที่ฉลาดกว่านี้ ชื่อว่า Union Find ถ้านิสิตสนใจสามารถศึกษาเพิ่มเติมเองได้)

คำแนะนำ 2 (ถ้ารู้วิธีทำแล้ว จะไม่อ่านก็ได้)

จากแนวคิด “ลิสต์ซ้อนลิสต์ของวงกลม” ในคำแนะนำ 1 สามารถสร้างคลาสของ “ลิสต์ซ้อนลิสต์ของวงกลม” ได้เป็นคลาส Cluster ดังนี้
นิสิตสามารถเรียกใช้ฟังก์ชัน build_clusters เพื่อสร้าง “ลิสต์ซ้อนลิสต์ของวงกลม” โดยจะอยู่ในรูปของ “ลิสต์ของ Cluster ของวงกลม”
และสามารถใช้เมทอดต่าง ๆ ที่มีให้เป็นประโยชน์ได้

```
class Cluster: # แทนแต่ละลิสต์ย่อยของวงกลม

    def __init__(self,c):
        self.circles = _____ # สร้างลิสต์ของวงกลมที่มีสมาชิกเป็นวงกลม c

    def add_circle(self,c):
        _____ # เพิ่มวงกลม c เข้าไปใน self.circles

    def merge_cluster(self,other):
        _____ # รวมลิสต์วงกลมจาก other เข้าไปใน self
        other.circles = []

    def is_empty(self): # ใช้ตรวจสอบว่า cluster นี้ว่างหรือไม่
        if _____:
            return True
        return False

    def contain_circle(self,c): # ใช้ตรวจสอบว่าวงกลม c ควรจะเข้าไปอยู่ใน cluster นี้หรือไม่
        for circle in self.circles:
            if c.touch(circle) != 'Free':
                return True
        return False

def build_clusters(Cs): # สร้างลิสต์ของ Cluster จากลิสต์ของวงกลม Cs
    clusters = []
    for i in range(len(Cs)):
        found_in_cluster = False
        active_cluster = -1
        for j in range(len(clusters)):
            if clusters[j].contain_circle(Cs[i]):
                if active_cluster == -1:
                    clusters[j].add_circle(Cs[i])
                    active_cluster = j
                else:
                    clusters[active_cluster].merge_cluster(clusters[j])
                    found_in_cluster = True
        if found_in_cluster == False:
            clusters.append(Cluster(Cs[i]))
    new_clusters = []
    for k in range(len(clusters)):
        if not clusters[k].is_empty():
            new_clusters.append(clusters[k])
    clusters = new_clusters
    return clusters
```

ข้อมูลนำเข้า

บรรทัดแรกแสดงจำนวนวงกลม (n)

บรรทัดที่สองตำแหน่งของจุด โดยแสดงเป็นจำนวนจริง 2 ค่า คือ พิกัด x,y

n บรรทัดที่เหลือ แสดงข้อมูลของวงกลมแต่ละวง ซึ่งประกอบด้วยจำนวนจริง 4 จำนวน คั่นด้วยเครื่องหมายเว้นวรรค แสดง พิกัดจุดศูนย์กลางวงกลมในแนวแกน x พิกัดจุดศูนย์กลางวงกลมในแนวแกน y รัศมี และความสว่างของสีของวงกลมนั้น

ข้อมูลส่งออก

แสดงค่าความสว่างสีของจุดคำถาม ถ้าจุดนี้ไม่อยู่ในหรือแตะวงกลมใด ๆ เลย ให้แสดงค่าเป็น 1.0

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
1 1.5 2.5 2.0 3.0 5.0 0.75	0.75
1 10.0 11.0 2.0 3.0 5.0 0.75	1.0
2 0.0 1.0 1.0 1.0 1.2 0.0 2.0 2.0 0.75 1.0	0.28089887640449435
3 5.5 9.8 1.0 1.0 1.2 0.0 2.0 2.0 0.75 1.0 5.0 10.0 2.5 0.8	0.8