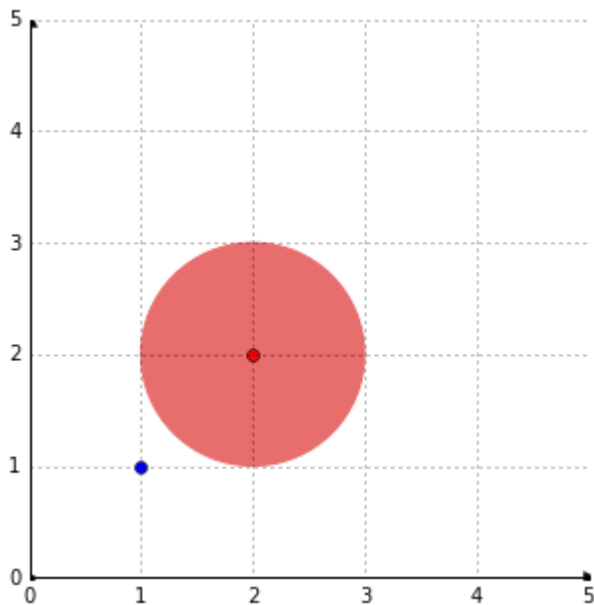


Cho tọa độ tâm và bán kính của một đường tròn cùng tọa độ của một điểm trên mặt phẳng 2 chiều

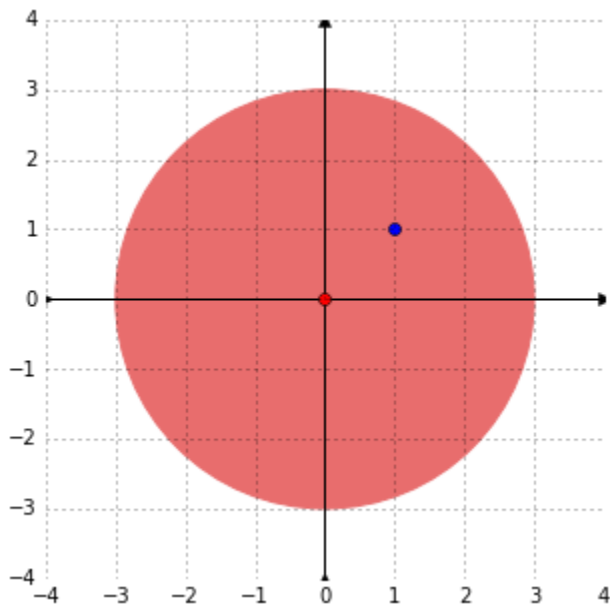
Kiểm tra xem điểm có nằm trong (hoặc nằm trên viền) của đường tròn hay không?

Ví dụ

- Với `point = [1, 1]`, `center = [2, 2]`, và `radius = 1`, kết quả `insideCircle(point, center, radius) = false`.



- Với `point = [1, 1]`, `center = [0, 0]`, và `radius = 3`, kết quả `insideCircle(point, center, radius) = true`.



Đầu vào/Đầu ra

- [Thời gian chạy] 0.5 giây

- **[Đầu vào] array.integer point**
Mảng 2 phần tử thể hiện tọa độ của điểm cần kiểm tra.
Điều kiện:
 $-100 \leq \text{point}[i] \leq 100.$
- **[Đầu vào] array.integer center**
Mảng 2 phần tử thể hiện tọa độ tâm của đường tròn
Điều kiện:
 $-10 \leq \text{center}[i] \leq 10.$
- **[Đầu vào] integer radius**
Bán kính đường tròn.
Điều kiện:
 $1 \leq \text{radius} \leq 5.$
- **[Đầu ra] boolean**
 true nếu point nằm trong đường tròn, false nếu ngược lại.

Lý thuyết :

- Cho hệ Oxy , 1 điểm $A(x, y)$ và 1 hình tròn có tâm tại $C(xC, yC)$, bán kính R
- Khi đó có 3 quan hệ :
 - $\text{dist}(A, C) < R$: điểm A nằm trong hình tròn
 - $\text{dist}(A, C) = R$: điểm A ở trên biên hình tròn
 - $\text{dist}(A, C) > R$: điểm A nằm ngoài hình tròn với $\text{dist}(A, C)$ là khoảng cách giữa điểm A và điểm C
- Code minh họa

```
int sqr(int x) { // tra ve binh phuong cua 1 so
    return x * x;
}

int checkPointCircle(vector <int> A, vector <int> C, int R) {
    int sqr_dist = sqr(C[0] - A[0]) + sqr(C[1] - A[1]);
    // binh phuong dist(A, C)
    if (sqr_dist < R * R) return -1; // trong hinh tron
    if (sqr_dist == R * R) return 0; // tren bien hinh tron
    if (sqr_dist > R * R) return 1; // ngoai hinh tron
}
```