```
// ---------------------------Basic Functions--------------------------
inline bool isEqual(double a, double b) {
	if(abs(a - b) <= EPS)
		return 1;
	return 0;
}

inline bool isGreater(double a, double b) {
	if(a >= b+EPS)
		return 1;
	return 0;
}

int Distance(int x1, int y1, int x2, int y2) {		// Without Sqrt
	int x = x1-x2;
	int y = y1-y2;
	return x*x + y*y;
}

double getAngle(double AB, double BC, double CA) {		// Returns Angle(Rad)
	return acos((AB*AB + BC*BC - CA*CA)/(2*AB*BC));
}

int PointToArea(int x1, int y1, int x2, int y2, int x3, int y3) {	// Returns Positive Area in if the points are clockwise, Negative for Anti-Clockwise
	return (x1*(y2-y3) + x2*(y3-y1) + x3*(y1-y2));			// Divide by 2 if Triangle area is needed
}

// -------------------------------Triangle-----------------------------
double TriangleArea(double AB, double BC, double CA) {
	double s = (AB + BC + CA)/2.0;
	return sqrt(s*(s-AB)*(s-BC)*(s-CA));
}

// -------------------------------Circle-------------------------------
struct circle {
	int x, y, r;
	circle(int _x, int _y, int _r) {
		x = _x;  y = _y;  r = _r;
	}
	double Area() {
		return PI*r*r;
	}
};

// Reference: https://www.mathsisfun.com/geometry/circle-sector-segment.html
double CircleSegmentArea(double r, double angle) {		// Circle Radius, Center Angle(Rad)
	return r * r * 0.5 * (angle - sin(angle));
}
double CircleSectorArea(double r, double angle) {		// Circle Radius, Center Angle(Rad)
	return r * r * 0.5 * angle;
}
double CircleArcLength(double r, double angle) {		// Circle Radius, Center Angle(Rad)
	return r * angle;
}

bool doIntersectCircle(circle c1, circle c2) {
	int dis = Distance(c1.x, c1.y, c2.x, c2.y);
	if(sqrt(dis) < c1.r+c2.r)   return 1;
```

```
    return 0;
}
```