





#include <iostream>

#include <cmath>

const double PI = 3.14159265358979323846;

struct Point {

double x, y;

};

Point calculatePosition(const Point& p1, double r1, double theta1, const Point& p2, double r2, double theta2) {

// Convert degrees to radians

double radTheta1 = theta1 \* PI / 180.0;

double radTheta2 = theta2 \* PI / 180.0;

// Calculate relative positions

double x1 = r1 \* cos(radTheta1);

double y1 = r1 \* sin(radTheta1);

double x2 = r2 \* cos(radTheta2);

double y2 = r2 \* sin(radTheta2);

// Calculate absolute positions

double absX1 = p1.x - x1;

double absY1 = p1.y - y1;

double absX2 = p2.x - x2;

double absY2 = p2.y - y2;

// Calculate own position (average position from both points)

double px = (absX1 + absX2) / 2;

double py = (absY1 + absY2) / 2;

return { px, py };

}

int main() {

// 定义已知点

Point P7 = { -30, 7 }; // P7 坐标

Point P8 = { 22, 0 }; // P8 坐标

// 输入案例数组

double cases[4][3] = {

{22, 0, 27.7}, // (P8 22 0) (P7 27.7 30)

{22, 0, 10.4}, // (P8 22 0) (P7 10.4 30)

{14, -30, 14.3}, // (P8 14 -30) (P7 14.3 0)

{45, 30, 76.65} // (P8 45 30) (P7 76.65 0)

};

// 测试每个输入案例

for (const auto& caseData : cases) {

double r1 = caseData[0]; // 距离 P7

double theta1 = caseData[1]; // 角度 P7

double r2 = caseData[2]; // 距离 P8

// 计算机器人的位置

Point pos = calculatePosition(P7, r1, theta1, P8, r2, 0.0); // theta2 设为 0

std::cout << "px=" << pos.x << ", py=" << pos.y << std::endl;

}

return 0;

}