```cpp


/*
 *              TRIANGULOS E CIRCULOS
 *
 */



struct circle{
    pv c;
    double r;
    circle(){c=pv(); r=0.0;}
    circle(pv a, double p) : c(a), r(p) {}
};



bool inside_circle(pv a, circle C){

    double c1_sq = (a.x - C.p.x)*(a.x - C.p.x);
    double c2_sq = (a.y - C.p.y)*(a.y - C.p.y);
    return (c1_sq + c2_sq) <= C.r*C.r;

}




double rCircumCircle(point a, point b, point c) {

    double ab = dist(a, b), bc = dist(b, c), ac = dist(a, c);


    double area = 0.5*cross(b-a, c-a);
    double ans =  (ab * bc * ac) / (4.0 * area);

    return ans;
}



bool center_circum_circle(pv o, pv p, pv q, pv &c){

    if(colinear(o, p, q)) return false;

    pv meio_op = o + ((p-o)*0.5);
    pv perp_op = perp_vec(p-o);
    line l1 = line(meio_op, perp_op);//mediatriz do segmento op


    pv meio_pq = p + ((q-p)*0.5);
    pv perp_pq = perp_vec(q-p);
    line l2 = line(meio_pq, perp_pq);//mediatriz do segmento pq


    line_intersection(l1, l2, c);
    return true;
}

```

```
61
62
63  bool center_in_circle(pv a, pv b, pv c, pv &ans){
64
65      if(colinear(a, b, c)) return false;
66
67      line l1 = bissetriz(a, b, c);
68      line l2 = bissetriz(b, c, a);
69
70      line_intersection(l1, l2, ans);
71      return true;
72  }
73
74
75
76
77  bool center_2ptsR(pv a, pv b, double r, pv &c1, pv &c2){
78
79      pv meio = a + ((b-a)*0.5);
80      pv p = unit_vec(perp_vec(b-a));
81      double d = dist(a, b),  aux = r*r - ((d*d)*0.25);
82
83      if(aux < 0.0) return false;//raio muito pequeno
84
85      //r² = d²/4.0 + h²
86
87
88      double h = sqrt( aux );
89      c1 = meio + (p*h);
90      c2 = meio + (p*(-h));
91      return true;
92  }
93
```