

```

104 Check and Find intersection of two segments (by points):
105 const double EPS = 1E-9;
106
107 struct pt {
108     double x, y;
109
110     bool operator<(const pt& p) const
111     {
112         return x < p.x - EPS || (abs(x - p.x) < EPS && y < p.y - EPS);
113     }
114 };
115
116 struct line {
117     double a, b, c;
118
119     line() {}
120     line(pt p, pt q)
121     {
122         a = p.y - q.y;
123         b = q.x - p.x;
124         c = -a * p.x - b * p.y;
125         norm();
126     }
127
128     void norm()
129     {
130         double z = sqrt(a * a + b * b);
131         if (abs(z) > EPS)
132             a /= z, b /= z, c /= z;
133     }
134
135     double dist(pt p) const { return a * p.x + b * p.y + c; }
136 };
137
138 double det(double a, double b, double c, double d)
139 {
140     return a * d - b * c;
141 }
142
143 inline bool betw(double l, double r, double x)
144 {
145     return min(l, r) ≤ x + EPS && x ≤ max(l, r) + EPS;
146 }
147
148 inline bool intersect_1d(double a, double b, double c, double d)
149 {
150     if (a > b)
151         swap(a, b);
152     if (c > d)
153         swap(c, d);
154     return max(a, c) ≤ min(b, d) + EPS;
155 }
156
157 //use this
158 //return ends points of intersection
159 //If the answer is a single point, the values written to left and right will be the
160 //same
161 bool intersect(pt a, pt b, pt c, pt d, pt& left, pt& right)
162 {
163     if (!intersect_1d(a.x, b.x, c.x, d.x) || !intersect_1d(a.y, b.y, c.y, d.y))
164         return false;
165     line m(a, b);

```