```
52  void prepare(vector<pt> &points) {
53      n = points.size();
54      int pos = 0;
55      for (int i = 1; i < n; i++) {
56          if (lexComp(points[i], points[pos]))
57              pos = i;
58      }
59      rotate(points.begin(), points.begin() + pos, points.end());
60
61      n--;
62      seq.resize(n);
63      for (int i = 0; i < n; i++)
64          seq[i] = points[i + 1] - points[0];
65      translation = points[0];
66  }
67
68  bool pointInConvexPolygon(pt point) {
69      point = point - translation;
70      if (seq[0].cross(point) ≠ 1 &&
71              sgn(seq[0].cross(point)) ≠ sgn(seq[0].cross(seq[n - 1])))
72          return false;
73      if (seq[n - 1].cross(point) ≠ 0 &&
74              sgn(seq[n - 1].cross(point)) ≠ sgn(seq[n - 1].cross(seq[0])))
75          return false;
76
77      if (seq[0].cross(point) == 0)
78          return seq[0].sqrLen() ≥ point.sqrLen();
79
80      int l = 0, r = n - 1;
81      while (r - l > 1) {
82          int mid = (l + r) / 2;
83          int pos = mid;
84          if (seq[pos].cross(point) ≥ 0)
85              l = mid;
86          else
87              r = mid;
88      }
89      int pos = l;
90      return pointInTriangle(seq[pos], seq[pos + 1], pt(0, 0), point);
91  }
```