```cpp
#include "../bits/stdc++.h"

using ld = long double;
using Point = std::complex<ld>;

const ld eps = 1e-9, pi = acos(-1.0);

namespace std
{
bool operator<(const Point &lhs, const Point &rhs)
{
  if (lhs.real() < rhs.real() - eps)
    return true;
  if (lhs.real() > rhs.real() + eps)
    return false;
  return lhs.imag() < rhs.imag();
}
} // namespace std

Point input_point()
{
  ld x, y;
  std::cin >> x >> y;
  return Point(x, y);
}

bool eq(ld a, ld b)
{
  return (abs(a - b) < eps);
}

ld dot(Point a, Point b)
{
  return real(conj(a) * b);
}

ld cross(Point a, Point b)
{
  return imag(conj(a) * b);
}

// CCW::counter clockwise
int ccw(Point a, Point b, Point c)
{
  b -= a;
  c -= a;
  if (cross(b, c) > eps)
    return 1; // a,b,c : counter-clockwise
  if (cross(b, c) < -eps)
    return -1; // a,b,c : clockwise
  if (dot(b, c) < 0)
    return 2; // c,a,b : on a line
  if (norm(b) < norm(c))
    return -2; // a,b,c : on a line
  return 0;   // a,c,b : on a line
}

```