```cpp
//
const double PI = acos(-1.0);
const double deg_rad = PI / 180.0;

class Qua {//Quaterion
public:
    Qua() {}

    Qua(double t, double x, double y, double z) : t(t), x(x), y(y), z(z) {}

    double t, x, y, z;

    Qua inv() {
        return Qua(t, -x, -y, -z);
    }

    friend Qua operator*(Qua l, Qua r) {
        Qua ans;
        ans.t = l.t * r.t - l.x * r.x - l.y * r.y - l.z * r.z;
        ans.x = l.t * r.x + l.x * r.t + l.y * r.z - l.z * r.y;
        ans.y = l.t * r.y + l.y * r.t + l.z * r.x - l.x * r.z;
        ans.z = l.t * r.z + l.z * r.t + l.x * r.y - l.y * r.x;
        return ans;
    }

};


class Point3 {
public:
    Point3() {}

    Point3(double x, double y, double z) : x(x), y(y), z(z) {}

    double x, y, z;

    double dis_to(Point3 b) {
        return sqrt((x - b.x) * (x - b.x) + (y - b.y) * (y - b.y) + (z - b.z) * (z - b.z));
    }

    double length() {
        return dis_to(Point3(0, 0, 0));
    }
```

```cpp

    friend Point3 operator+(Point3 l, Point3 r) {
        return Point3(l.x + r.x, l.y + r.y, l.z + r.z);
    };

    friend Point3 operator-(Point3 l, Point3 r) {
        return Point3(l.x - r.x, l.y - r.y, l.z - r.z);
    };

    Point3 operator*(double b) {
        return Point3(x * b, y * b, z * b);
    }

    friend double operator*(Point3 l, Point3 r) {
        return l.x * r.x + l.y * r.y + l.z * r.z;
    };

    //叉乘
    Point3 cross(Point3 r) {
        return Point3(y * r.z - z * r.y, z * r.x - x * r.z, x * r.y - y * r.x);
    }

    Point3 unit() {
        double len = length();
        return Point3(x / len, y / len, z / len);
    }

    void debug() {
        printf("(%lf,\t%lf,\t%lf)\n", x, y, z);
    }

};

class Segment {
public:

    Point3 u, v;

    Segment() {};

    Segment(Point3 u, Point3 v) : u(u), v(v) {};

    double dis_to(Point3 x) {
        if (((x - u) * (v - u)) * ((x - v) * (v - u)) <= 0) {
            return ((x - u).cross(v - u)).length() / (v - u).length();
        } else {
            return min(x.dis_to(u), x.dis_to(v));
```

```
 91         }
 92     }
 93
 94     double dis_to(Segment b){
 95         if((v-u).cross(b.v-b.u).length()<=EPS){
 96
 97         }
 98     }
 99
100 };
101
102 class Point3r {
103 public:
104     Point3r() {}
105
106     Point3r(double phi, double theta, double r) : phi(phi)
    , theta(theta), r(r) {}
107
108     double phi, theta, r;//x-axis[-pi,pi] z-axis[0,pi]
    radius
109
110     Point3 to_xyz() {
111         Point3 re;
112         re.z = cos(theta) * r;
113         double xy = sin(theta) * r;
114         re.x = cos(phi) * xy;
115         re.y = sin(phi) * xy;
116         return re;
117     }
118 };
119
120 class Point3jw {
121 public:
122     Point3jw() {}
123
124     Point3jw(double lo, double la, double r) : lo(lo), la(
    la), r(r) {}
125
126     double lo, la, r;//longitude[-180,180] latitude[-90,90
    ] radius
127
128     Point3 to_xyz() {
129         Point3 re;
130         re.z = cos((90 - la) * deg_rad) * r;
131         double xy = sin((90 - la) * deg_rad) * r;
132         re.x = cos(lo * deg_rad) * xy;
133         re.y = sin(lo * deg_rad) * xy;
134         return re;
135     }
136 };
```