

2D vector template:

```
struct p //point;
{
    double x,y;
    p(double x=0,double y=0)
    {
        this->x=x;
        this->y=y;
    }
};

p mv(p a,p b) //make vector
{
    return p(b.x-a.x,b.y-a.y);
}

double dp(p a,p b) //dot product
{
    return a.x*b.x+a.y*b.y;
}

double cp(p a,p b) //cross product
{
    return a.x*b.y-a.y*b.x;
}

p vr(p v,double an) //vector rotation an radious angle
{
    return p(v.x*cos(an)-v.y*sin(an),v.x*sin(an)+v.y*cos(an));
}

p add(p a,p b) //two vector add
{
    return p(a.x+b.x,a.y+b.y);
}

double value(p a) //value of a vector
{
    return sqrt(dp(a,a));
}

p lv(p a,double l) //l th time of a vector
{
    double v=value(a);
    return p((a.x*l)/v,(a.y*l)/v);
}
```

```

p pointseg(p a,p b,p c) //min distance point of a,b segment from point c
{
    if(dp(mv(a,b),mv(a,c))<0) return a;
    if(dp(mv(b,a),mv(b,c))<0) return b;
    double l=(dp(mv(a,b),mv(a,c)))/value(mv(a,b));
    return add(a,lv(mv(a,b),l));
}

double dis(p a,p b) //distance of two point
{
    return sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y));
}

bool onseg(p a,p b,p c) //if a,b,c colinear check if c lies on a,b segment
{
    return (c.x>=min(a.x,b.x) && c.x<=max(a.x,b.x) &&
            c.y>=min(a.y,b.y) && c.y<=max(a.y,b.y));
}

bool inseg(p a,p b,p c,p d) //check if a,b and c,d segment intersect
{
    double d1,d2,d3,d4;
    d1=cp(mv(a,b),mv(a,c));
    d2=cp(mv(a,b),mv(a,d));
    d3=cp(mv(c,d),mv(c,a));
    d4=cp(mv(c,d),mv(c,b));

    if(d1*d2<0 && d3*d4<0) return true;

    if(d1==0 && onseg(a,b,c)) return true;
    if(d2==0 && onseg(a,b,d)) return true;
    if(d3==0 && onseg(c,d,a)) return true;
    if(d4==0 && onseg(c,d,b)) return true;

    return false;
}

p px(p a,p b,double x) // point on line ab at 0<x<1 time of full distance
{
    return p(a.x+(b.x-a.x)*x,a.y+(b.y-a.y)*x);
}

```

3D vector template:

```
struct p //point;
{
    double x,y,z;
    p(double x=0,double y=0,double z=0)
    {
        this->x=x;
        this->y=y;
        this->z=z;
    }
};

p mv(p a,p b) //make vector
{
    return p(b.x-a.x,b.y-a.y,b.z-a.z);
}

double dp(p a,p b) //dot product
{
    return a.x*b.x+a.y*b.y+a.z*b.z;
}

double value(p a) //value of a vector
{
    return sqrt(dp(a,a));
}

double cp(p a,p b) //cross product
{
    return value(p(a.y*b.z-b.y*a.z,a.z*b.x-a.x*b.z,a.x*b.y-a.y*b.x));
}

p add(p a,p b) //two vector add
{
    return p(a.x+b.x,a.y+b.y,a.z+b.z);
}

p lv(p a,double l) //l th time of a vector
{
    double v=value(a);
    return p((a.x*l)/v,(a.y*l)/v,(a.z*l)/v);
}

p pointseg(p a,p b,p c) //min distance point of a,b segment from point c
{
    if(dp(mv(a,b),mv(a,c))<0) return a;
    if(dp(mv(b,a),mv(b,c))<0) return b;
    double l=(dp(mv(a,b),mv(a,c)))/value(mv(a,b));
    return add(a,lv(mv(a,b),l));
}

double dis(p a,p b) //distance of two point
{
    return sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y)+(a.z-b.z)*(a.z-b.z));
}
```