点

const double eps = 1e-10;

int dcmp (double x) {

if(fabs(x)<eps) return 0;

return x < 0? -1: 1;

}

struct Point{

double x, y;

Point (double x=0, double y=0):x(x), y(y) {}

Point operator+ (const Point &p)

{return Point (x+ p. x, y + p. y);}

减法: {return Point (x - p. x, y - p. y);}

Point operator\*(double p) {return Point (x\*p, y\*p);}

除法: {return Point (x/p, y/p);}

};

typedef Point Vector;

注: Vector简写为V

**注: 点积 叉积 长度 角度**

double Dot (V a, V b) {return a. x\*b. x + a. y\*b. y;}

double Cross (V a, V b) {return a. x\*b. y - a. y\*b. x;}

double Length (V a) {return sqrt (Dot (a, a));}

double Angle (V a, V b)

{return acos (Dot (a, b)/Length(a)/Length(b));}

**注: 向量旋转 x’=xcosa-ysina, y’=xsina+ycosa**

Vector Rotate (V A, double rad) {return

Vector (A. x\*cos(rad) - A. y\*sin(rad),

A. x\*sin(rad) + A. y\*cos(rad));

}