**//Convex Points**

**/\***

**2D points in point array (distinct points), parameter total numbers of points.**

**return convex points in a stack;**

**\*/**

struct Z{

long long x,y;

};

Z point[100000+10];

double IsLeft(Z a, Z b, Z c)

{

double x=(b.x-a.x)\*(b.y-c.y)-(b.y-a.y)\*(b.x-c.x);

return x;

}

double Distance(Z a, Z b)

{

double x=(a.x-b.x)\*(a.x-b.x)+(a.y-b.y)\*(a.y-b.y);

x=sqrt(x); return x;

}

bool Compare(Z a, Z b)

{

double d1,d2;

d1=IsLeft(point[0],a,b);

if(d1>0) return true;

if(d1<0) return false;

d1=Distance(point[0],a);

d2=Distance(point[0],b);

if(d1>d2) return true;

else return false;

}

double Angle(Z a, Z b, Z c)

{

double x,A,B,C;

A=Distance(b,c);

B=Distance(c,a);

C=Distance(a,b);

x=acos((C\*C+A\*A-B\*B)/(2\*C\*A))\*180.0/PI; return x;

}

stack<Z> convexPoints(int n)

{

int mn=0;

Z lft, mddl;

for(int i=1;i<n;i++)

if(point[i].y<point[mn].y || (point[i].y==point[mn].y && point[i].x<point[mn].x))

mn=i;

swap(point[mn],point[0]);

sort(point+1,point+n,Compare);

stack<Z>stk;

stk.push(point[0]);

stk.push(point[1]);

for(int j=2;j<n;j++)

{

while(stk.size()>1)

{

mddl=stk.top();

stk.pop();

lft=stk.top();

double xx=IsLeft(lft, mddl, point[j]);

if(xx<0.0) continue;

stk.push(mddl);

stk.push(point[j]);

break;

}

}

return stk;

}