#include

#include

#include

using namespace std;

#define eps (1e-9)

const double pi = 2\*acos(0.0);

int n, m;

double r;

struct P

{

double x, y;

} d[510], c;

struct interval

{

double c;

bool open;

} g[2010];

bool operator < (interval a, interval b)

{

return a.c+eps < b.c || b.c+eps >= a.c && a.open && !b.open;

}

void angles(P p, double r, double &a, double &b)

{

double ac = acos(sqrt(p.x\*p.x+p.y\*p.y)/(2\*r));

if (p.x >= -eps && p.x <= eps)

{

if (p.y > 0)

{

a = pi/2-ac;

b = pi/2+ac;

return;

}

a = 3\*pi/2-ac;

b = 3\*pi/2+ac;

return;

}

double at = atan(p.y/p.x);

if (p.x > eps)

{

a = at-ac;

b = at+ac;

return;

}

a = pi+at-ac;

b = pi+at+ac;

}

bool solve(P p, double r)

{

double a, b;

int l = 0, i;

for (i = 0; i < n; i++)

{

d[i].x -= p.x;

d[i].y -= p.y;

if ((fabs(d[i].x) >= eps || fabs(d[i].y) >= eps) && d[i].x\*d[i].x+d[i].y\*d[i].y < 4\*r\*r+eps)

{

angles(d[i], r, a, b);

if (a < eps)

a += 2\*pi;

if (a > 2\*pi+eps)

a -= 2\*pi;

if (b < eps)

b += 2\*pi;

if (b > 2\*pi+eps)

b -= 2\*pi;

if (a < b+eps)

{

g[l].c = a;

g[l++].open = 1;

g[l].c = b;

g[l++].open = 0;

}

else

{

g[l].c = a;

g[l++].open = 1;

g[l].c = b+2\*pi;

g[l++].open = 0;

}

}

d[i].x += p.x;

d[i].y += p.y;

}

if (l < 2\*m)

return 0;

int k;

sort(g, g+l);

for (k = i = 0; i < l; i++)

{

if (g[i].open)

k++;

else

k--;

if (k >= m)

{

c.x = r\*cos(g[i].c)+p.x;

c.y = r\*sin(g[i].c)+p.y;

return 1;

}

}

return 0;

}

int main()

{

int t, i;

double l, k;

scanf("%d%d", &n, &m);

for (i = 0; i < n; i++)

scanf("%lf%lf", &d[i].x, &d[i].y);

m--;

l = 0.0;

k = 7100;

for (t = 0; t < 30; t++)

{

r = (l+k)/2.0;

for (i = 0; i < n && !solve(d[i], r); i++);

if (i == n)

l = r;

else

k = r;

}

printf("%.8lf\n%.8lf %.8lf\n", r, c.x, c.y);

return 0;

}