/\*

TASK:msweeper

LANG:C++

\*/

#define PROG1 1

#define TEST 0

#define PROG2 0

#if PROG2

#include

#include

#include

using namespace std;

#define inf 1000000000

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

int n, r;

struct P

{

int x, y;

} c[10010], st[20010], p;

bool operator < (P a, P b)

{

if (b.x == p.x && b.y == p.y)

return 0;

if (a.x == p.x && a.y == p.y)

return 1;

double p1, p2;

if (a.x > 0)

p1 = atan(a.y\*1.0/a.x);

if (a.x == 0)

if (a.y > 0)

p1 = pi/2.0;

else

p1 = -pi/2.0;

if (a.x < 0)

p1 = pi-atan(b.y\*1.0/b.x);

p1 -= acos(r/sqrt((a.x-p.x)\*(a.x-p.x)+(a.y-p.y)\*(a.y-p.y)+0.0));

if (b.x > 0)

p2 = atan(b.y\*1.0/b.x);

if (b.x == 0)

if (b.y > 0)

p2 = pi/2.0;

else

p2 = -pi/2.0;

if (b.x < 0)

p2 = pi-atan(b.y\*1.0/b.x);

p2 -= acos(r/sqrt((b.x-p.x)\*(b.x-p.x)+(b.y-p.y)\*(b.y-p.y)+0.0));

return p1+eps < p2;

}

int main()

{

double l = 0.0;

int m = 0, d, i;

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

p.x = p.y = inf;

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

{

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

if (p.y > c[i].y || p.y == c[i].y && p.x > c[i].x)

p = c[i];

}

sort(c, c+n);

st[m++] = c[0];

st[m++] = c[1];

for (i = 2; i < n; i++)

{

while (m > 1)

{

d = (st[m-2].x-st[m-1].x)\*(c[i].y-st[m-1].y)-(st[m-2].y-st[m-1].y)\*(c[i].x-st[m-1].x);

if (d < 0 || d == 0 &&

(c[i].x-st[m-2].x)\*(c[i].x-st[m-2].x)+(c[i].y-st[m-2].y)\*(c[i].y-st[m-2].y) >

(st[m-1].x-st[m-2].x)\*(st[m-1].x-st[m-2].x)+(st[m-1].y-st[m-2].y)\*(st[m-1].y-st[m-2].y))

break;

}

}

return 0;

}

#endif

#if PROG1

#include

#include

#define inf 2000000000

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

int r;

bool t[10010];

struct P

{

int x, y;

} c[10010], ans[10010], p;

int main()

{

freopen("input.txt", "r", stdin);

double l = 0.0;

int n, m, k, d, i;

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

if (n == 1)

{

printf("%lf\n", 2\*pi\*r);

return 0;

}

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

{

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

if (c[k].y > c[i].y || c[k].y == c[i].y && c[k].x > c[i].x)

k = i;

}

ans[m++] = c[k];

while (m <= 1 || ans[0].x != ans[m-1].x || ans[0].y != ans[m-1].y)

{

k = -1;

p = ans[m-1];

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

if (!t[i])

if (k == -1)

k = i;

else

{

d = (c[i].x-p.x)\*(c[k].y-p.y)-(c[i].y-p.y)\*(c[k].x-p.x);

if (d > 0 || d == 0 && (c[i].x-p.x)\*(c[i].x-p.x)+(c[i].y-p.y)\*(c[i].y-p.y) >

(c[k].x-p.x)\*(c[k].x-p.x)+(c[k].y-p.y)\*(c[k].y-p.y))

k = i;

}

t[k] = 1;

ans[m++] = c[k];

l += sqrt((c[k].x-p.x)\*(c[k].x-p.x)+(c[k].y-p.y)\*(c[k].y-p.y)+0.0);

}

printf("%lf\n", l+2\*pi\*r);

return 0;

}

#endif

#if TEST

#include

#include

#include

#include

int main()

{

freopen("input.txt", "w", stdout);

int n, r, i;

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

printf("%d %d\n", n, r);

srand(time(0));

for (i = -n; i <= n; i+=4)

if (i)

{

printf("%d %d\n", i, (int)(sqrt(n\*n-i\*i+0.0))+rand()%5000);

printf("%d %d\n", i, -(int)(sqrt(n\*n-i\*i+0.0))-rand()%5000);

}

return 0;

}

#endif