

```

/*矩阵并的面积和周长*/
#include <cstdio>
#include <algorithm>
#include <map>
#include <set>
#include <queue>
using namespace std;
/*

```

边界为 1

内部为 2

当边界无效时为 3

矩阵的周长就是偶数行奇数列的 1 的宽度 和 奇数列偶数行的高度

矩阵的面积就是里面 2 的面积计算方法： $(r-1) * (d-u)$

```

0 0 0 0 0 0
0 0 0 0 0 0
0 0 1 1 1 0
0 0 1 2 1 0
0 0 1 1 1 0
0 0 0 0 0 0
*/
struct B {
    double x1, y1, x2, y2;
    void init() {
        scanf("%lf%lf%lf%lf", &x1, &y1, &x2, &y2);
        if(x1 > x2) swap(x1, x2);
        if(y1 > y2) swap(y1, y2);
    };
};

int const inf = 0x3f3f3f3f, maxn = 20100;
int x1, y1, x2, y2, n, mx, my;
int m[maxn][maxn];
set<double> x, y;
set<double>::iterator si;
map<double, int> hx, hy;
map<int, double> hhx, hhy;
B b[maxn];

```

```

double gets() {
    double ans = 0;
    for(int i=3; i<mx; i+=2)
        for(int j=3; j<my; j+=2)
            if(m[i][j]==2)
                ans += (hhx[i+1]-hhx[i-1])*(hhy[j+1]-hhy[j-1]);
    return ans;
}

```

```

double getL() {
    double ans=0;
    for(int i=2; i<mx; i+=2)
        for(int j=3; j<my; j+=2)
            if(m[i][j]==1)
                ans += hhy[j+1]-hhy[j-1];
    for(int i=3; i<mx; i+=2)
        for(int j=2; j<my; j+=2)
            if(m[i][j]==1)
                ans += hhx[i+1]-hhx[i-1];
    return ans;
}

```

```

int main() {
    const bool debug = false;
    int i, j, k, cs=1;
    while(~scanf("%d", &n), n) {

        x.clear(); y.clear();
        for(i = 0; i < n; i++) {
            b[i].init();
            x.insert(b[i].x1); x.insert(b[i].x2);
            y.insert(b[i].y1); y.insert(b[i].y2);
        }
        hx.clear(); hy.clear();

```

//把地图扩大二倍后，矩阵内部就可以被填充，矩阵边界就可以走了

//对 x 离散化

```

for(si=x.begin(), mx=2; si!=x.end();
    hx[*si]=mx, hhx[mx] = *si, si++, mx+=2) ;

```

//对 y 离散化

```

for(si=y.begin(), my=2; si!=y.end();
    hhy[*si]=my, hhy[my] = *si, si++, my+=2);

```

```

        //初始化矩阵
        for(i = 0; i < mx; ++i) {
            fill(m[i], m[i] + my, 0);
        }

    //填充矩阵，填充为 1
    for(i = 0; i < n; i++) {
        int xuper = hx[b[i].x2];
        int yuper = hy[b[i].y2];

        //填充上下边界
        for(j = hx[b[i].x1]; j <= xuper; j++) {
            if(m[j][hy[b[i].y1]]==0)m[j][hy[b[i].y1]]=1;
            if(m[j][hy[b[i].y2]]==0)m[j][hy[b[i].y2]]=1;
        }

        //填充左右边界
        for(k = hy[b[i].y1]; k <= yuper; k++) {
            if(m[hx[b[i].x1]][k]==0)m[hx[b[i].x1]][k]=1;
            if(m[hx[b[i].x2]][k]==0)m[hx[b[i].x2]][k]=1;
        }

        //填充矩阵内部
        for(j = hx[b[i].x1] + 1; j < xuper; j++)
            for(k = hy[b[i].y1] + 1; k < yuper; k++)
                m[j][k]=2;
    }

    //此处已不属于周长，标记为 3
    for(i=1;i<mx-1;i++)
        for(j=1;j<my-1;j++)
            if(m[i][j] ==1 && m[i-1][j]&& m[i][j-1]
                && m[i+1][j]&& m[i][j+1]) m[i][j]=3;
    double S=getS();
    printf("area: %.2f\n\n",cs++,S);

    int L = (int)getL();
    printf("length:%d\n",L);
    }
    return 0;
}

```