题意：

给出数字n和k，n表示接下来将输入n个在x轴上的闭区间[li,ri]，找出被覆盖了至少k次的点，并求由这些点组成的连续区间的数目，并使该数目最小。输出该数目并将区间从左到右（x的正方向）输出。

用到了扫描线的思想。

先将所有的线段端点排序，这样相当于把线段排了序，对于重合的点应该是左端点优先，用pair可以简洁地做到这一点。

用一条竖直线，从左到右的把整个区间扫描一遍。

每次遇到一个左端点，覆盖数+1，遇到一个右端点，覆盖数-1。

我们称“覆盖数的变化”为 “事件”。

在这道题里，一条线段对应的是两个事件。

我们定义一个event，记录下每条线段的L和时间的种类——即“入”或“出”。

将“入”记为+1，“出”为-1是比较容易理解的。

在这道题里我们可以发现，事件的发生顺序是由L决定的，所以我们按照L的升序将线段排序。

然后在扫描的过程中我们就可以得到 当覆盖数为k时的区间。

注意pair的第二个元素左要小于右端点，因为左端点优先。

pair会自动按照第一个元素优先，第二个元素其次排序。

#include<bits/stdc++.h>

#include <iostream>

#include <stdio.h>

#include<cstring>

using namespace std;

typedef long long ll;

vector<pair<int,int> > v;

vector<pair<int,int> > solution;

int main()

{

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

int n,k,l,r;

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

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

{

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

v.push\_back(make\_pair(l,0));

v.push\_back(make\_pair(r,1));

}

sort(v.begin(),v.end());

ll ans(0);

for(int i=0;i<v.size();i++)

{

if(v[i].second==0)

{

ans++;

if(ans==k)//注意不要写成>=，只有当等于的时候才是这个点满足条件和不满足条件的分界，才应该放入集合中

solution.push\_back(v[i]);

}

else

{

if(ans==k)

solution.push\_back(v[i]);

ans--;

}

}

printf("%d\n",solution.size()/2);//个数一定是偶数的

for(int i=0;i<solution.size()/2;i++)

{

printf("%d %d\n",solution[2\*i].first,solution[2\*i+1].first);

}

return 0;

}