题意：有一些化合物，每个化合物都由两种元素组成的（每个元素用一个大写字母表示）。你是一个装箱的工人，从实验员那里按照顺序依次把一些简单化合物装到车上。但这里存在一个安全隐患：如果车上存在k个简单的化合物，正好包含k中元素，那么它们将组成一个易爆的混合物。为了安全起见，每当你拿到一个化合物时，如果它和已装的化合物形成易爆混合物，你就应当拒绝装车；否则就应该装车。编程输出有多少个没有装车的化合物。

怎么样k个化合物k种元素？（怎么样k条边，k个点）只有闭合环的时候

我们把每个元素看成顶点，则一个简单化合物就是一条边。当整个图存在环的时候，组成环的边对应的化合物是危险的，反之则是安全的

这样，我们可以用一个并查集来维护图的连通分量，每次得到一个简单化合物（x，y）时检查x和y是否在同一个集合中。如果是（这样就能连成环了），则拒绝，反之则接受

你读入的一行的两个是组成化合物的两个，已经就是自动隐形的连成边了

#include<iostream>

#define maxn 100010

using namespace std;

int a[maxn];

int find(int x)

{

return a[x]!=x?a[x]=find(a[x]):x;//带路径压缩

}

int main()

{

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

int x,y;

while(cin>>x)

{

for(int i=1;i<=maxn;i++)a[i]=i;

int refuse=0;

while(x!=-1)

{

cin>>y;

x=find(x);y=find(y);

if(x==y) refuse++;else a[x]=y;

cin>>x;

}

cout<<refuse<<endl;

}

return 0;

}

这个三目运算符很值得学习

赋值语句的值就是赋的值

同时实现了给a[x]赋值find(a[x])

没有用启发式合并