

I like kicking you!

遍历字符串，找到字符 'k'，则往后遍历 3 位，看是否匹配 "kick"，匹配则 $count++$ ，并且题目说明可重叠，那么下一个字符的开始则跳到当前找到的这个 "kick" 的最后一位 'k'，最终输出 $count$

I like cat!

并查集问题，但这题我们需要记录一个 $bottom$ 数组和一个 $next$ 数组，假设 v 并入 u ， $bottom$ 数组用来记录 v 应该并入 u 的哪个位置， $next$ 数组用来记录当前点的下一个点为多少编号的点，当 v 要并入 u 时， $next[bottom[u]] = v$ ，即 u 这个组合遍历完后应该访问 v 这个组合， $bottom[u] = bottom[v]$ 即修改并入位置，因为 u 与 v 已经位于同一组合，因此应该公用同一个被其它点接入的位置，同时进行正常并查集操作。

代码：

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
ll f[150005],nxt[150005],bottom[150005];
ll find(ll x)
{
    return x==f[x]? x:f[x] = find(f[x]);
}

int main(){
    ll n;
    cin>>n;

    for(ll i=1;i<=n;i++) {
        bottom[i]=i;
        f[i]=i;
        nxt[i]=0;
    }
    ll x,y;
    for(ll i=1;i<=n-1;i++){
        cin>>x>>y;
        ll f1=find(x);
        ll f2=find(y);
        nxt[bottom[f1]]=f2;
        bottom[f1]=bottom[f2];
        f[f2]=f1;
    }

    ll st=find(f[1]);
    for(ll i=st;i;i=nxt[i]) cout<<i<<" ";
    cout<<endl;
}
```