

//求两个字符串中长度大于等于K的公共子串的总个数

```
#include <iostream>
#include <cstring>
#include <algorithm>
#include <cmath>
using namespace std;
const int maxn=200010;
int s[maxn],sa[maxn],rank[maxn],t[maxn],t2[maxn],height[maxn],c[maxn];
int sta[maxn][2];
void build_sa(int n,int m){
    int i,*x=t,*y=t2;
    //初始化基数排序
    for (i=0;i<m;i++) c[i]=0;
    for (i=0;i<n;++i) ++c[ x[i]=s[i] ];
    for (i=1;i<m;++i) c[i]+=c[i-1];
    for (i=n-1;i>=0;--i) sa[ --c[x[i]] ]=i;
    //倍增基数排序
    for (int k=1;k<=n;k<=<=1){
        int p=0;
        //先根据sa数组直接排序第二关键词
        for (i=n-k;i<n;++i) y[p++]=i;
        for (i=0;i<n;++i) if (sa[i]>=k) y[p++]=sa[i]-k;
        //再排序第一关键词
        for (i=0;i<m;i++) c[i]=0;
        for (i=0;i<n;++i) ++c[x[y[i]]];
        for (i=1;i<m;++i) c[i]+=c[i-1];
        for (i=n-1;i>=0;--i) sa[--c[x[y[i]]] ]=y[i];
        //根据sa数组重新构造x数组
        swap(x,y);
        p=1;x[sa[0]]=0;
        for (i=1;i<n;++i)
            x[sa[i]]= y[sa[i-1]]==y[sa[i]] && y[sa[i-1]+k]==y[sa[i]+k] ? p-1:p++;
        if (p>=n) break;
        m=p;
    }
}
void getHeight(int n){
    int k=0;
    for (int i=0;i<n;++i) rank[sa[i]]=i;
    for (int i=0;i<n;++i) {
        if (k) k--;
        if (rank[i]){
            int j=sa[rank[i]-1];
            while(s[i+k]==s[j+k]) k++;
        }
        height[rank[i]]=k;
    }
}
int main(){
    //freopen("datain.txt","r",stdin);
    ios::sync_with_stdio(false);
    cin.tie(0);
    int k;
    while(cin>>k && k){
        //读入
        string s1,s2;
```

```

cin>>s1>>s2;
int len1=s1.length(),n=0;
for (int i=0;s1[i];i++) s[n++]=s1[i]-'A'+2;
s[n++]=1;
for (int i=0;s2[i];i++) s[n++]=s2[i]-'A'+2;
s[n++]=0;
//构造数组
build_sa(n,70);
getHeight(n);
//预处理
for (int i=0;i<n;i++) height[i]=max(0,height[i]-k+1);
//sa[i-1]在第一个字符串中, sa[i]在第二个字符串中
long long ans=0,tot=0,top=0;
for (int i=1;i<n;i++){
    if (height[i]<=0){
        top=0;tot=0;
    }else{
        int cnt=0;
        if (sa[i-1]<len1){ cnt++;tot+=height[i];
        }
        while(top && height[i]<=sta[top-1][0]){
            top--;
            tot+=(height[i]-sta[top][0])*sta[top][1];
            cnt+=sta[top][1];
        }
        sta[top][0]=height[i];
        sta[top++][1]=cnt;
        if (sa[i]>len1) ans+=tot;
    }
}
//sa[i-1]在第二个字符串中, sa[i]在第一个字符串中
top=0;tot=0;
for (int i=1;i<n;i++){
    if (height[i]<=0){
        top=0;tot=0;
    }else{
        int cnt=0;
        if (sa[i-1]>len1){ cnt++;tot+=height[i];
        }
        while(top && height[i]<=sta[top-1][0]){
            top--;
            tot+=(height[i]-sta[top][0])*sta[top][1];
            cnt+=sta[top][1];
        }
        sta[top][0]=height[i];
        sta[top++][1]=cnt;
        if (sa[i]<len1) ans+=tot;
    }
}
//输出
cout<<ans<<endl;

```

```

}

```

```

}

```