const int N=2e6+5;

int ch[N][26],l[N],fa[N],cnt,root,last;

char s[N];

void init(){last=root=++cnt;}

void insert(int x){//字符的位置

int p=last,np=++cnt,c=s[x]-'a';l[np]=x;last=np;

for (;p && !ch[p][c];p=fa[p])ch[p][c]=np;

if (!p)fa[np]=root;

else {

int q=ch[p][c];

if (l[p]+1==l[q])fa[np]=q;

else {

int nq=++cnt;l[nq]=l[p]+1;

memcpy(ch[nq],ch[q],sizeof(ch[q]));

fa[nq]=fa[q];fa[np]=fa[q]=nq;

for (;ch[p][c]==q;p=fa[p])ch[p][c]=nq;

}

}

}

const long long mod=998244353;

long long pw(long long x,long long y)

{

y=(y%(mod-1)+mod-1)%(mod-1);

long long res=1;

while(y>0)

{

if(y&1) res=res\*x%mod;

x=x\*x%mod;

y>>=1;

}

return res;

}

void NTT(long long \*A,int len,int opr)

{

for(int i=1,t=0;i<len;i++)

{

int k=len;

do t^=(k>>=1); while(~t&k);

if(i<t) swap(A[i],A[t]);

}

for(int h=2;h<=len;h<<=1)

{

long long wn=pw(3,(mod-1)/h\*opr);

for(int i=0;i<len;i+=h)

{

long long w=1;

for(int j=i;j<i+(h>>1);j++)

{

long long t1=A[j],t2=A[j+(h>>1)]\*w%mod;

A[j]=(t1+t2)%mod;

A[j+(h>>1)]=(t1-t2+mod)%mod;

w=w\*wn%mod;

}

}

}

if(opr<0)

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

A[i]=(A[i]\*((1-mod)/len)%mod+mod)%mod;

}

typedef complex<double> cd;

typedef vector<cd> vcd;

void fft(vcd &A,int k){

int As=A.size();

if(As==1) return ;

vcd B0(As/2),B1(As/2);

for(int i=0;i<As;i++) if(i&1) B1[i/2]=A[i];else B0[i/2]=A[i];

fft(B0,k);fft(B1,k);

cd i(0,1),wn(exp((k\*2\*acos(-1)/(double)As)\*i)),w(1,0);

for(int i=0;i<As/2;i++){

A[i]=B0[i]+w\*B1[i];

A[i+As/2]=B0[i]-w\*B1[i];

w=w\*wn;

}

}

