https://ac.nowcoder.com/acm/contest/7607/C

给定字符串T首尾相接无限循环，Q次询问，每次给定一字符串S，求从T开头开始，需要向后跳多少位才能顺序经过S中的所有字符(S中存在压缩的单一重复字符)

如果每次都从1开始，i个字符$'a'$，显然前缀和一下，算出一次循环中$'a'$出现的次数，可以得到位于整块中的$'a'$，剩下的在前缀和中二分查找一下即可

考虑上一次结束的位置位于一次循环的第$last$位，我们还是边读入边算出位于整块中的$'a'$，然后尝试使用$last->len$中的$'a'$，如果不够，就令答案加上这一段的长度，从1开始再找；如果够的话二分查找到具体的位置

可以先忽略$last->n$这一段直接算位于整块中的$'a'$，是因为即使$last->len$中的$'a'$够多出现了空余，那么后面所有整块的$'a'$向前移动补齐后，$last$向后移动若干整块，在整块中的相对位置没有变化

```
#include <iostream>
#include <cstdio>
#include <cstring>
#define int long long
using namespace std;
const int MAX=1e6+5;
const int MAXp=30;
const int mod=998244353;
char a[MAX],b[MAX];
inline int mo(int x){return (x%mod+mod)%mod;}
int pre[MAXp][MAX];
int LEN,len,q;
int last,unit,tmpans;   //last上次位置；unit表示一整个字符串有多少所需字符；num表示经过tmpans轮整字符串后还剩的字符数量
signed main(){
    //freopen("T3.in","r",stdin);
    scanf("%s",a+1);
    LEN=strlen(a+1);
    for(int i=1;i<=LEN;i++){
        for(int j=0;j<=26;j++){
            pre[j][i]=pre[j][i-1];
        }
        pre[a[i]-'a'][i]++,pre[26][i]++;
    }
    scanf("%lld",&q);
    while(q--){
```

```
        bool flag=true;
        scanf("%s",b+1);
        len=strlen(b+1);
        int r=0,ans=0;
        last=1;
        for(int cur,i=1;i<=len;i=r+1){
            if(b[i]=='*')cur=26;
            else cur=b[i]-'a';
            unit=pre[cur][LEN];
            if(!unit){flag=false;break;}
            int num=1;
            tmpans=0,r=i;
            if(i+1<=len && isdigit(b[i+1])){
                num=0;
                while(r+1<=len && isdigit(b[r+1])){
                    r++;
                    num=num*10+b[r]-'0';
                    tmpans=mo(tmpans*10+num/unit);
                    num%=unit;
                }
                tmpans=mo(tmpans*LEN);
                if(!num)num=unit,tmpans=mo(tmpans-LEN);
            }
            if(num>pre[cur][LEN]-pre[cur][last-1])num-=(pre[cur][LEN]-
pre[cur][last-1]),tmpans=mo(tmpans+LEN-last+1),last=1;
            int tmp=lower_bound(pre[cur]+last,pre[cur]+LEN+1,pre[cur][last-
1]+num)-pre[cur];
            tmpans=mo(tmpans+tmp-last+1),last=tmp+1,ans=mo(ans+tmpans);
            //cout<<ans<<" ";
        }
        if(!flag)printf("-1\n");
        else printf("%lld\n",ans);
    }
    return 0;
}
```