

//AC自动机总模板

```
#include <iostream>
#include <queue>
#include <map>
#include <cstring>
#include <cstdio>
using namespace std;
const int sigma_size=26;
const int maxnode=1e5+5;
struct AhoCorasickAutomata{
    int ch[maxnode][sigma_size];
    int val[maxnode],nxt[maxnode],last[maxnode];
    int sz;
    int idx(char c) {
        return c-'a';
    }

    void init(){
        sz=1;
        memset(ch[0],0,sizeof(ch[0]));
    }

    void print(int j){
        if (j){
            printf("%d: %d\n",j,val[j]);
            print(last[j]);
        }
    }

    void insert(char *s,int len,int v){
        int u=0;
        for (int i=0;i<len;i++){
            int c=idx(s[i]);
            if (!ch[u][c]){
                memset(ch[sz],0,sizeof(ch[sz]));
                val[sz]=0;
                ch[u][c]=sz++;
            }
            u=ch[u][c];
        }
        val[u]=v;
    }

    void getFail(){
        queue <int> q;
        nxt[0]=0;
        //初始化
        for (int c=0;c<sigma_size;c++){
            int u=ch[0][c];
            if (u) {
                nxt[u]=0;
                last[u]=0;
                q.push(u);
            }
        }
        while(!q.empty()){
            int r=q.front();
            int p=nxt[r];
            while(p){
                last[r]=val[p];
                r=p;
            }
            p=nxt[r];
            while(p){
                last[r]=val[p];
                r=p;
            }
            q.push(nxt[r]);
        }
    }
};
```

```

        q.pop();
        for (int c=0;c<sigma_size;c++){
            int u=ch[r][c];
            if (!u) {
                ch[r][c]=ch[ nxt[r] ][c];
                continue;
            }
            q.push(u);
            int v=nxt[r];
            while(v && !ch[v][c]) v=nxt[v];
            nxt[u]=ch[v][c];
            last[u]=val[ nxt[u] ] ? nxt[u] : last[ nxt[u] ] ;
        }
    }
}

void find(char *T,int len){
    int j=0;
    for (int i=0;i<len;i++){
        int c=idx(T[i]);
        while(j && !ch[j][c]) j=nxt[j];
        j=ch[j][c];
        if (val[j]) print(j);
        else if (last[j]) print(last[j]);
    }
}

}ac;
char text[1000001];
int main(){
    int n;
    while(~scanf("%d",&n) && n){
        ac.init();
        char P[151][80];
        for (int i=1;i<=n;i++){
            scanf("%s",P[i]);
            ac.insert(P[i],strlen(P[i]),i);
        }
        ac.getFail();
        scanf("%s",text);
        ac.find(text,strlen(text));
    }
    return 0;
}

```