A

```
const int maxn=1000010;
char str[maxn];
char s[maxn*2];
int p[maxn*2];
int len;
void manacher(){
    int i;
    s[0]='$';
    s[1]='#';
    len=strlen(str);
    for(i=0;i<len;i++){
        s[i*2+2]=str[i];
        s[i*2+3]='#';
    }
    len=i*2+2;
    s[len]='\0';
    int maxl,id=0;
    maxl=0;
    mem0(p);
    for(int i=0 ;i<len ;i++){
        if(p[id]+id>i) p[i]=min(p[2*id-i],p[id]+id-i);
        else p[i]=1;
        while(s[i+p[i]]==s[i-p[i]]) p[i]++;
        if(p[i]+i>p[id]+id) id=i;
        if(p[i]-1>maxl) maxl=p[i]-1;
    }
    cout<<maxl<<endl;
}
int main(){
    while(~scanf("%s",str)){
        manacher();
    }
}
```

B

```
int main(){
    int T;
    cin>>T;
    priority_queue<ll,vector<ll>,greater<ll> >q;
    while(T--){
        int n;
        Sca(n);
        for(int i=0;i<n;i++){
            int s;
            Sca(s);
            q.push(s);
        }
        ll ans=0,x,y;
        while(!q.empty()){
            x=q.top();
            q.pop();
```

```
            if(q.empty()) break;
            y=q.top(),q.pop();
            ans+=x+y;
            q.push(x+y);
        }
        Prill(ans);
        printf("\n");
        if(T!=0) printf("\n");
    }
}
```

C

```
#include<stdio.h>
#include<iostream>
#include<algorithm>
#define N 10000005
#define int long long
using namespace std;
struct node{int v,sum,x,dep;}A[N],B[N];
bool operator<(node a,node b)
{
    if(a.v==b.v)return a.dep<b.dep;
    return a.v<b.v;
}
int n,k,l1=1,r1,l2=1,r2;
main()
{
    int i,j;node tmp;
    scanf("%lld%lld",&n,&k);
    for(i=1;i<=n;i++)
    {
        scanf("%lld",&A[++r1].v);
        A[r1].x=A[r1].v;
    }
    if((n-1)%(k-1))r1+=k-1-(n-1)%(k-1);
    sort(A+l1,A+r1+1);
    while(r1-l1+r2-l2+2>1)
    {
        i=0;tmp.sum=tmp.v=tmp.x=tmp.dep=0;
        while(i<k)
        {
            if(l1>r1)
            {
                tmp.v+=B[l2].v;
                tmp.sum+=B[l2].sum+B[l2].x;
                tmp.x+=B[l2].x;
                tmp.dep=max(tmp.dep,B[l2].dep+1);
                l2++;
            }
            else if(l2>r2)
            {
                tmp.v+=A[l1].v;
                tmp.sum+=A[l1].sum+A[l1].x;
                tmp.x+=A[l1].x;
                tmp.dep=max(tmp.dep,A[l1].dep+1);
                l1++;
```

```
            }
            else if(A[l1]<B[l2])
            {
                tmp.v+=A[l1].v;
                tmp.sum+=A[l1].sum+A[l1].x;
                tmp.x+=A[l1].x;
                tmp.dep=max(tmp.dep,A[l1].dep+1);
                l1++;
            }
            else
            {
                tmp.v+=B[l2].v;
                tmp.sum+=B[l2].sum+B[l2].x;
                tmp.x+=B[l2].x;
                tmp.dep=max(tmp.dep,B[l2].dep+1);
                l2++;
            }
            i++;
        }
        B[++r2]=tmp;
    }
    printf("%lld\n%lld",B[l2].sum,B[l2].dep);
}
```

D

```
const int M = 26;

struct node{
    char s[15];
    bool operator < (const node &a) const {
        if(strcmp(s, a.s) <= 0) return 1;
        else return 0;
    }
};
vector<node> d[M];
int main(){
    for(int i = 0; i < 26; ++ i) d[i].clear();
    char s[25];
    node temp;
    while(gets(temp.s), temp.s[0] != '\0'){
        d[temp.s[0]-'a'].push_back(temp);
    while(scanf("%s", s) == 1){
        int ans = 0, j;
        int sss = s[0]-'a';
        for(int i = 0; i < d[sss].size(); ++ i){
            for(j = 0; j < strlen(s); ++ j){
                if(s[j] != d[sss][i].s[j]) break;
            }
            if(j == strlen(s)) ++ans;
        }
        printf("%d\n", ans);
    }
    return 0;
}
```

E

```
const ull P = (ll)1e10 + 7;
const ull MOD = 998244353;
char s[maxn];
ull PP[maxn];
ull pre_Hash[maxn], sux_Hash[maxn];
ull Hash1(int l, int r) {
    return (pre_Hash[r] - pre_Hash[l - 1] * PP[r - l + 1] % MOD + MOD) % MOD;
}
ull Hash2(int l, int r) {
    return (sux_Hash[l] - sux_Hash[r + 1] * PP[r - l + 1] % MOD + MOD) % MOD;
}

void init() {
    PP[0] = 1;
    for (int i = 1; i < maxn; i++) {
        PP[i] = PP[i - 1] * P % MOD;
    }
}
int main() {
    init();
    int t;
    cin >> t;
    while (t--) {
        int ans = 0, len;
        cin >> s + 1;
        len = strlen(s + 1);
        pre_Hash[0] = 0, sux_Hash[len + 1] = 0;
        for (int i = 1; i <= len; i++)
            pre_Hash[i] = (pre_Hash[i - 1] * P + (s[i] - 'a' + 1)) % MOD;
        for (int i = len; i >= 1; i--)
            sux_Hash[i] = (sux_Hash[i + 1] * P + (s[i] - 'a' + 1)) % MOD;
        int x = 0;
        int one_hs = 0, two_hs = 0;
        while (s[x + 1] == s[len - x] && x + 1 < len - x)
            x++;
        for (int i = x + 1; i <= len - x; i++)
            if (Hash1(x + 1, i) == Hash2(x + 1, i))
                if (i - x > one_hs) {
                    one_hs = i - x;
                    two_hs = 0;
                }
        for (int i = len - x; i >= x + 1; i--)
            if (Hash1(i, len - x) == Hash2(i, len - x))
                if (len - x - i + 1 > one_hs) {
                    one_hs = len - x - i + 1;
                    two_hs = 1;
                }
        for (int i = 1; i <= x; i++)
            cout << s[i];

        if (two_hs == 0)
            for (int i = x + 1; i <= x + one_hs; i++)
                cout << s[i];

        else
```

```
            for (int i = len - x; i >= len - x - one_hs + 1; i--)
                cout << s[i];
        for (int i = len - x + 1; i <= len; i++)
            cout << s[i];
        cout << endl;
    }
}
```

F

```
#include <bits/stdc++.h>
#define ll long long
using namespace std;


const int MOD = 1e9+7;
const int MAXN = 6*1e5+5;
const int X = 257;

int n, m;
char ch[MAXN];
set <ll> s;
ll f[MAXN];

void init() {
    f[0] = 1;
    for(int i = 1;i < MAXN; i++) {
        f[i] = f[i-1]*X % MOD;
    }
}
ll Hash(char *ch) {
    int len = strlen(ch);
    ll tmp = 0;
    for(int i = 0;i < len; i++) {
        tmp = (tmp*X + ch[i])%MOD;
    }
    return tmp;
}

bool charge(char *ch) {
    int len = strlen(ch);
    ll h = Hash(ch);
    for(int i = 0;i < len; i++)
        for(ll c = 'a'; c <= 'c'; c++) {
            if(c == ch[i]) continue;
            if(s.find((((c-ch[i])*f[len-i-1] + h)%MOD+MOD)%MOD) != s.end())
                return true;
        }
    return false;
}

int main() {
    init();
    scanf("%d%d", &n, &m);
    for(int i = 0;i < n; i++) {
        scanf("%s", ch);
        s.insert(Hash(ch));
```

```
    }
    for(int i = 0;i < m; i++) {
        scanf("%s", ch);
        puts(charge(ch) ? "YES":"NO");
    }
    return 0;
}
```