一

```
int getMaxElement(const node* f){
    return f?max(f->val,getMaxElement(f->next)):INT_MIN;
}
int size(const node* f){
    return f?1+size(f->next):0;
}
double average(const node* f,long long sum=0,int size=0){
    return f?average(f->next,sum+f->val,size+1):(double)sum/size;
}
```

二

```
#include<cstdio>
const int maxn=1e6+5;
int n,A[maxn],pos[maxn];
int stk[maxn],top;
bool instk[maxn];
void solve(){
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d",&A[i]);
        pos[A[i]]=i;
        instk[i]=0;
    }
    top=0;
    int x;
    int cur=0;
    for(int i=0;i<n;i++){
        scanf("%d",&x);
        while(cur<=pos[x]){
            stk[++top]=A[cur];
            instk[A[cur]]=top;
            cur++;
        }
        if(stk[top]!=x)return puts("NO"),void();
        top--;
        instk[x]=0;
    }puts("YES");
}
int main(){
```

```
    solve();
    return 0;
}
```

三

```
#include<cstdio>
using namespace std;
const int N=9;
int oper[N+5];
void dfs(int cur,int sum){
    if(cur>N){
        if(sum==110){
            for(int i=1;i<=N;i++){
                if(oper[i]&&i>1)putchar(~oper[i]?'+':'-');
                printf("%d",i);
            }puts("=110");
        }return;
    }
    int res=0;
    int ccur=cur;
    while(cur<=N){
        res=res*10+cur;
        oper[ccur]=1;dfs(cur+1,sum+res);
        if(ccur>1)oper[ccur]=-1;dfs(cur+1,sum-res);
        oper[cur]=0;
        cur++;
    }oper[ccur]=0;
}
int main(){
    dfs(1,0);
    return 0;
}
```