```
#include <iostream>

using namespace std;

int a[] = {1,2,3,4};

void reverse(int a[],int l,int r){
    while(l<r){
        // 交换两个变量
        int temp = a[l];
        a[l] = a[r];
        a[r] = temp;
        l ++;
        r --;
    }
}

void show(int a[],int len){
    for(int i  = 0;i<len;++i)
    cout<<a[i];
}

void loopR(int a[],int l,int r,int k){
    //reverse all
    reverse(a,l,r);
    //reverse k-th
    reverse(a,l,l+k-1);
    //reverse tail
    reverse(a,l+k,r);
}

void deleteItem(int a[],int l,int r,int num){
    int i = l;
    for( ; i <= r;++i)
        if(a[i]==num) break;
    for( ;i<=r;++i){
        a[i-1] = a[i];
    }
}

int main(){
    loopR(a,0,3,3);
    show(a,4);
    return 0;
}
```

```
#include <iostream>

using namespace std;

struct node{
    int val;
    node* next;
};

node* create(){
    int n;
    cin>>n;
    if(n==-1) return NULL;
    node * ret = new node;
    ret->val  = n;
    ret->next = create();
    return ret;
}

void show(node * head){
    while(head){
        cout<<head->val<<" -> ";
        head =head ->next;
    }
    cout<<"NULL"<<endl;
}

node* reverse(node* head){
    if(head == NULL) return NULL;
    node * p = head ->next;
    head ->next = NULL;
    while(p){
        node * t = p->next;
        p->next = head;
        head = p;
        p = t;
    }
    return head;
}
node * merge(node * p,node* q){
    node* fake = new node;
    fake -> next = NULL;
    node * t = fake;
    while(p&&q){
        node * minp ;
        if(p->val < q->val){
            minp = p;
            p = p->next;
        }
        else{
            minp = q;
            q = q->next;
        }
        node* temp = minp->next;
        minp->next = t->next;
        t->next = minp;
        t = minp;
    }
    if(p) t->next = p;
    if(q) t->next = q;
    return fake->next;
}

int main(){
    node * head = create();
    show(head);
    node * head1 = create();
    show(head1);

    head = merge(head,head1);
    show(head);
    return 0;
}
```