//1.cpp

#include<iostream>

using namespace std;

class node{

    public:

    int v;

    node \*nxt;

};

class list{

    public:

    node \*head;

    node \*tail;

    list(){

        head=NULL;

        tail=NULL;

    }

    ~list(){

        node \*tmp;

        while(head != NULL){

            tmp = head->nxt;

            delete head;

            head=tmp;

        }

    }

    void attach(node \*pnn){

        if(tail!=NULL)

            tail->nxt=pnn;

        tail=pnn;

        if(head==NULL)

            head=pnn;

    }

    void disp(){

        node \*trav = head;

        while(trav!=NULL){

            cout << trav->v << " ";

            trav=trav->nxt;

        }

        cout << endl;

    }

};

void remove\_equal(list \*l1,list \*l2){

    node \*t1 = l1->head,\*t2 = l2->head,\*prev=NULL,\*nxt;

    while(t1!=NULL){

        nxt = t2->nxt;

        if(t1->v == t2->v){

            prev->nxt = t2->nxt;

            delete t2;

        }else{

            prev = t2;

        }

        t1 = t1->nxt,t2 = nxt;

    }

}

int main(){

    list \*l1 = new list,\*l2 = new list;

    node \*pnn;

    int n;

    cout << "enter number of nodes: ";

    cin >> n;

    cout << "list 1" << endl;

    for(int i=0;i<n;i++){

        pnn = new node;

        cin >> pnn->v;

        pnn->nxt = NULL;

        l1->attach(pnn);

    }

    cout << "list 2" << endl;

    for(int i=0;i<n;i++){

        pnn = new node;

        cin >> pnn->v;

        pnn->nxt = NULL;

        l2->attach(pnn);

    }

    remove\_equal(l1,l2);

    cout << "list 2: ";

    l2->disp();

    delete l1;

    delete l2;

    return 0;

}

//2.cpp

#include<iostream>

using namespace std;

class node{

    public:

    int v;

    node \*nxt;

};

class list{

    public:

    node \*head;

    node \*tail;

    list(){

        head=NULL;

        tail=NULL;

    }

    ~list(){

        node \*tmp;

        while(head != NULL){

            tmp = head->nxt;

            delete head;

            head=tmp;

        }

    }

    void attach(node \*pnn){

        if(tail!=NULL)

            tail->nxt=pnn;

        tail=pnn;

        if(head==NULL)

            head=pnn;

    }

    void disp(){

        node \*trav = head;

        while(trav!=NULL){

            cout << trav->v << " ";

            trav=trav->nxt;

        }

        cout << endl;

    }

};

void total\_between(list \*l,int st,int en){

    node \*trav = l->head;

    int total=0,flag=0;

    while(trav->v != en && trav!=NULL){

        if(flag==1){

            total += trav->v;

        }

        if(trav->v == st)

            flag=1;

        trav = trav->nxt;

    }

    if(trav == NULL)//error

        cout << "end value not found" << endl;

    else

        cout << "total " << st << ":" << en << " = " << total << endl;

}

int main(){

    list \*l1 = new list,\*l2 = new list;

    node \*pnn;

    int n;

    cout << "list 1 number of nodes: ";

    cin >> n;

    for(int i=0;i<n;i++){

        pnn = new node;

        cin >> pnn->v;

        pnn->nxt = NULL;

        l1->attach(pnn);

    }

    cout << "list 2 number of nodes: ";

    cin >> n;

    for(int i=0;i<n;i++){

        pnn = new node;

        cin >> pnn->v;

        pnn->nxt = NULL;

        l2->attach(pnn);

    }

    node \*prev,\*trav=l1->head;

    int i=0;

    while(trav!=NULL){

        if(i%2){

            total\_between(l2,prev->v,trav->v);

        }

        prev = trav;

        trav = trav->nxt;

        i++;

    }

    delete l1;

    delete l2;

    return 0;

}

//3.cpp

#include<iostream>

using namespace std;

class node{

    public:

    int v;

    node \*nxt;

};

class list{

    public:

    node \*head;

    node \*tail;

    list(){

        head=NULL;

        tail=NULL;

    }

    ~list(){

        node \*tmp;

        while(head != NULL){

            tmp = head->nxt;

            delete head;

            head=tmp;

        }

    }

    void attach(node \*pnn){

        if(tail!=NULL)

            tail->nxt=pnn;

        tail=pnn;

        if(head==NULL)

            head=pnn;

    }

    void disp(){

        node \*trav = head;

        while(trav!=NULL){

            cout << trav->v << " ";

            trav=trav->nxt;

        }

        cout << endl;

    }

};

void cut(list \*l,list \*newL){

    node \*trav,\*start=NULL;

    trav = l->head;

    int ishead=0;

    if(l->head->v < 0)

        ishead=1;

    while(trav!=NULL){

        if(ishead==1 && trav->nxt->v < 0)

            break;

        if(start==NULL && trav->nxt->v < 0)

            start=trav;

        else if(trav->v < 0 && trav != start->nxt)

            break;

        trav = trav->nxt;

    }

    if(ishead){

        if(newL->head==NULL){

            newL->head = l->head;

            newL->tail = trav;

        }

        else{

            newL->tail->nxt = l->head;

            newL->tail = trav;

        }

        if(trav == l->tail){

            l->head = NULL;

            l->tail = NULL;

        }else{

            l->head=trav->nxt;

            trav->nxt = NULL;

        }

    }else{

        if(newL->head==NULL){

            newL->head = start->nxt;

            newL->tail = trav;

        }else{

            newL->tail->nxt = start->nxt;

            newL->tail = trav;

        }

        start->nxt = trav->nxt;

        trav->nxt=NULL;

        if(trav == l->tail)

            l->tail= start;

    }

}

int main(){

    int n;

    list \*\*l = new list\*[20],\*newL = new list;

    node \*pnn;

    for(int i=0;i<20;i++){

        l[i] = new list;

        cout << "list " << i << " number: ";

        cin >> n;

        for(int j=0;j<n;j++){

            pnn = new node;

            cin >> pnn->v;

            pnn->nxt = NULL;

            l[i]->attach(pnn);

        }

    }

    for(int i=0,j=19;i<10;i++,j--){

        cut(l[i],newL);

        cut(l[j],newL);

    }

    cout << "newL :";

    newL->disp();

    for(int i=0;i<20;i++)

        delete l[i];

    delete []l;

    delete newL;

    return 0;

}

//4.cpp

#include<iostream>

using namespace std;

class node{

    public:

    int v;

    node \*nxt;

};

class list{

    public:

    node \*head;

    node \*tail;

    list(){

        head=NULL;

        tail=NULL;

    }

    ~list(){

        node \*tmp;

        while(head != NULL){

            tmp = head->nxt;

            delete head;

            head=tmp;

        }

    }

    void attach(node \*pnn){

        if(tail!=NULL)

            tail->nxt=pnn;

        tail=pnn;

        if(head==NULL)

            head=pnn;

    }

    void disp(){

        node \*trav = head;

        while(trav!=NULL){

            cout << trav->v << " ";

            trav=trav->nxt;

        }

        cout << endl;

    }

};

int main(){

    int len=30;

    // cout << "number of lists: ";

    // cin >> len;

    list \*\*l = new list\*[len];

    node \*pnn,\*trav,\*trav2,\*no;

    int n;

    for(int i=0;i<len;i++){

        l[i] = new list;

        cout << "list " << i << " number of nodes: ";

        cin >> n;

        for(int j=0;j<n;j++){

            pnn = new node;

            cin >> pnn->v;

            pnn->nxt = NULL;

            l[i]->attach(pnn);

        }

    }

    int ct,j;

    for(int i=0;i<len;i+=2){

        ct=0,j=0;

        trav = l[i]->head;

        trav2 = l[i+1]->head;

        if(trav->v == -1){

            l[i+1]->head = trav2->nxt;

            trav2->nxt = trav->nxt;

            trav->nxt = trav2;

            continue;

        }

        while(trav!=NULL){

            if(trav->nxt->v==-1)

                break;

            ct++;

            trav=trav->nxt;

        }

        while(trav2!=NULL){

            if(j==ct)

                break;

            j++;

            trav2=trav2->nxt;

        }

        if(trav2==NULL){

            cout << "index out of range" << endl;

            continue;

        }

        no = trav2->nxt;

        trav2->nxt = trav2->nxt->nxt;

        if(no == l[i+1]->tail)

            l[i+1]->tail = trav2;

        no->nxt = trav->nxt->nxt;

        trav->nxt->nxt = no;

    }

    for(int i=0;i<len;i++){

        l[i]->disp();

    }

    for(int i=0;i<len;i++){

        delete l[i];

    }

    delete l;

    return 0;

}