#include <iostream>

using namespace std;

struct Node

{

struct Node \*next;

int value;

};

Node\* NewNode(int value)

{

Node \*node = new Node();

node->next = NULL;

node->value = value;

return node;

}

Node\* CreateList(int \*num, int n)

{

if (!num || n <= 0) return NULL;

Node \*head = NewNode(num[0]);

Node \*tail = head;

for (int i = 1; i < n; i++)

{

Node \*node = NewNode(num[i]);

tail->next = node;

tail = node;

}

return head;

}

ListNode\* MergeList(ListNode \*list1, ListNode \*list2)

{

if (!list1) return list2;

if (!list2) return list1;

ListNode \*head, \*tail, \*p = list1, \*q = list2;

head = tail = p->val > q->val ? p : q;

head == p ? p = p->next : q = q->next;

while (p && q)

{

ListNode \*k = p->val < q->val ? q : p;

tail->next = k;

tail = k;

p->val < q->val ? q = q->next : p = p->next;

}

if (p) tail->next = p;

if (q) tail->next = q;

return head;

}

void ShowList(Node \*head)

{

if (!head) return;

for (Node \*p = head; p != NULL; p = p->next)

{

cout << p->value << endl;

}

}

int main(int argc, const char \* argv[]) {

int num1[] = { 7, 6, 5, 4, 3, 2 };

int num2[] = { 8, 6, 4, 2, 0, -2 };

Node \*list1 = CreateList(num1, sizeof(num1) / sizeof(int));

Node \*list2 = CreateList(num2, sizeof(num2) / sizeof(int));

Node \*head = MergeList(list1, list2);

ShowList(head);

return 0;

}

