1.

释放由内存，防止内存泄漏。

3.

void reverse() {

node\* p = nullptr;

node\* u;

node\* l = head;

while (l != nullptr) {

u = l;

l = l->suc;

p = u;

}

head = p;

}

4.

void newList(nodeList A,nodeList B) {

node\* j = A.head->next;

node\* k = B.head->next;

nodeList C;

while (j != A.rear || k != B.rear) {

if (j != A.rear && ((j->data) >= (k->data) || k == B.rear)) {

C.insert(j->data);

j = j->next;

}

if (k != B.rear && ((k->data) >= (j->data)|| j == A.rear)) {

C.insert(k->data);

k = k->next;

}

}

C.reverse();

}

5.

bool conclude(nodeList list) { //list为双向链表

node\* h = list.head;

node\* r = list.rear;

while (l != h) {

if (h->data != r->data)

return false;

else {

h = h->next

r = r->pre;

}

}

return true;

}