2.12

//函数实现A,B两表连接，最后为循环链表A；

void TowTry(Stu \*a, Stu \*&b)//B的头指针可能要置空

{

Stu \*p = NULL;

Stu \*q = NULL;

//如若B表为空

if (b->Next == NULL){

delete p;//删除头结点

b = NULL;//将指针置空

return;

}

else{//B表非空

//A表遍历到最后一个元素

q = a;

while (q ->Next != a){

q = q->Next;

}

q ->Next = b->Next;//A的尾巴，连接到了B的首元结点

//遍历B表到最后一个元素

p = b;

while (p->Next != b){

p = p->Next;

}

//把B的尾巴连到A的头上

p->Next = a;

//最后释放B的头结点

Delete b;

b = NULL;

}

}

2.23

bool IsRight(Stu \*a)

{

Stu \*first, \*last;

first = a->Next;

last = a->Prior;

if (last->Prior == first)//只有首元结点

{

return true;

}

while (first != last){

//前后同时遍历对比数值

if (first->date != last->date) return false;

else{

first = first->Next;

last = last->Prior;

}

}

return true;

}

2.24

//函数实现逆置头结点双循环链表

bool TurnList(Stu \*a)

{

Stu \*p, \*q, \*l;

//用类似单链表的头插法来实现这个过程

if (a->Next->Next == a) return true;

else if (a->Next == a) return false;

p = a->Next;

q = a->Next->Next;

l = a->Prior;

p->Next = a;

p->Prior = a;

p = q;

q = q->Next;

while (a->Next != l){

p->Prior = a;

p->Next = a->Next;

a->Next = p;

p = q;

q = q->Next;

}

return true;

}