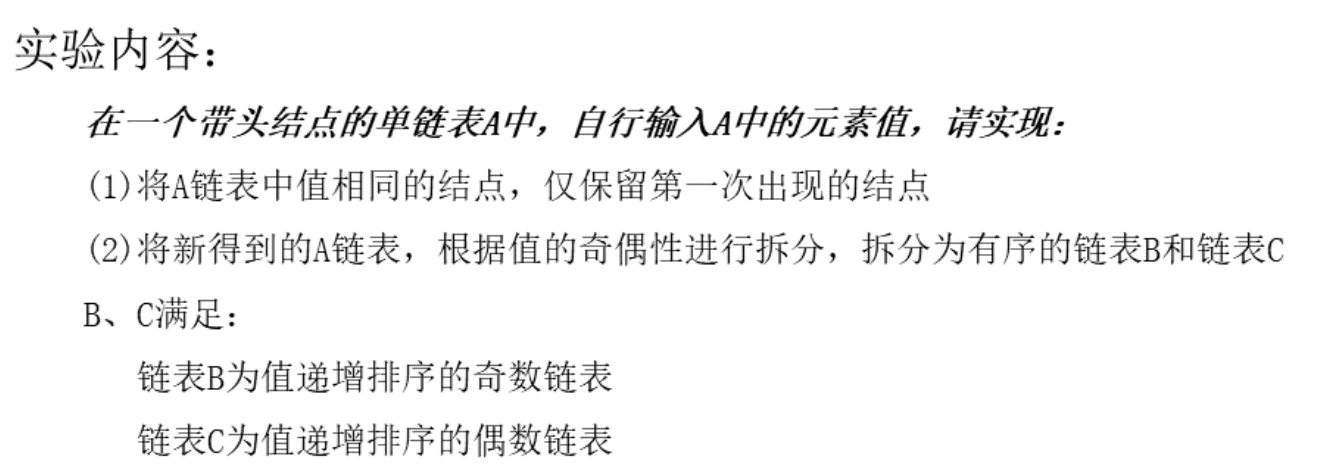
数据结构实验二

班级：20信安法 学号：2013484 姓名：张世伟

1. 实验内容



1. 设计思路
2. 设计链表结构；
3. 设计函数Insert实现链表的升序插入；
4. 设计Delete递归函数,删除链表中重复的节点；
5. 设计函数Divide，遍历删除节点后的链表，根据链表节点是否为奇数，调用Insert函数将节点插入新建的两个链表中；
6. 设计Output函数输出链表。
7. 核心代码
8. Insert函数

node\* Insert(node\* head, int data){

node\* p = head;

node\* new\_node = new node;

new\_node->val = data;

if (p->val > data){

new\_node->val = head->val;

new\_node->next = head->next;

head->next = new\_node;

head->val = data;

}

else{

node\* q = head->next;

while (q && data > q->val){

p = q;

q = q->next;

}

new\_node->next = q;

p->next = new\_node;

}

return head;

}

1. Delete递归函数

node\* Delete(node\* head){

node\* p, \* temp = head;

if (!head->next)

return head;

head->next = Delete(head->next);

p = head->next;

while{

if (head->val == p->val){

temp->next = p->next;

free(p);

p = temp->next;

}

else {

p = p->next;

temp = temp->next;

}

}

return head;

}

1. Divide函数

void Divide(node\* head){

node\* head2, \* head3;

head2 = new node;

head3 = new node;

for (node\* i = head; i; i = i->next){

if (!i->val % 2 ) {

Insert(head3, i->val);

}

else{

Insert(head2, i->val);

}

}

cout << endl;

cout << "B : ";

Output(head2->next);

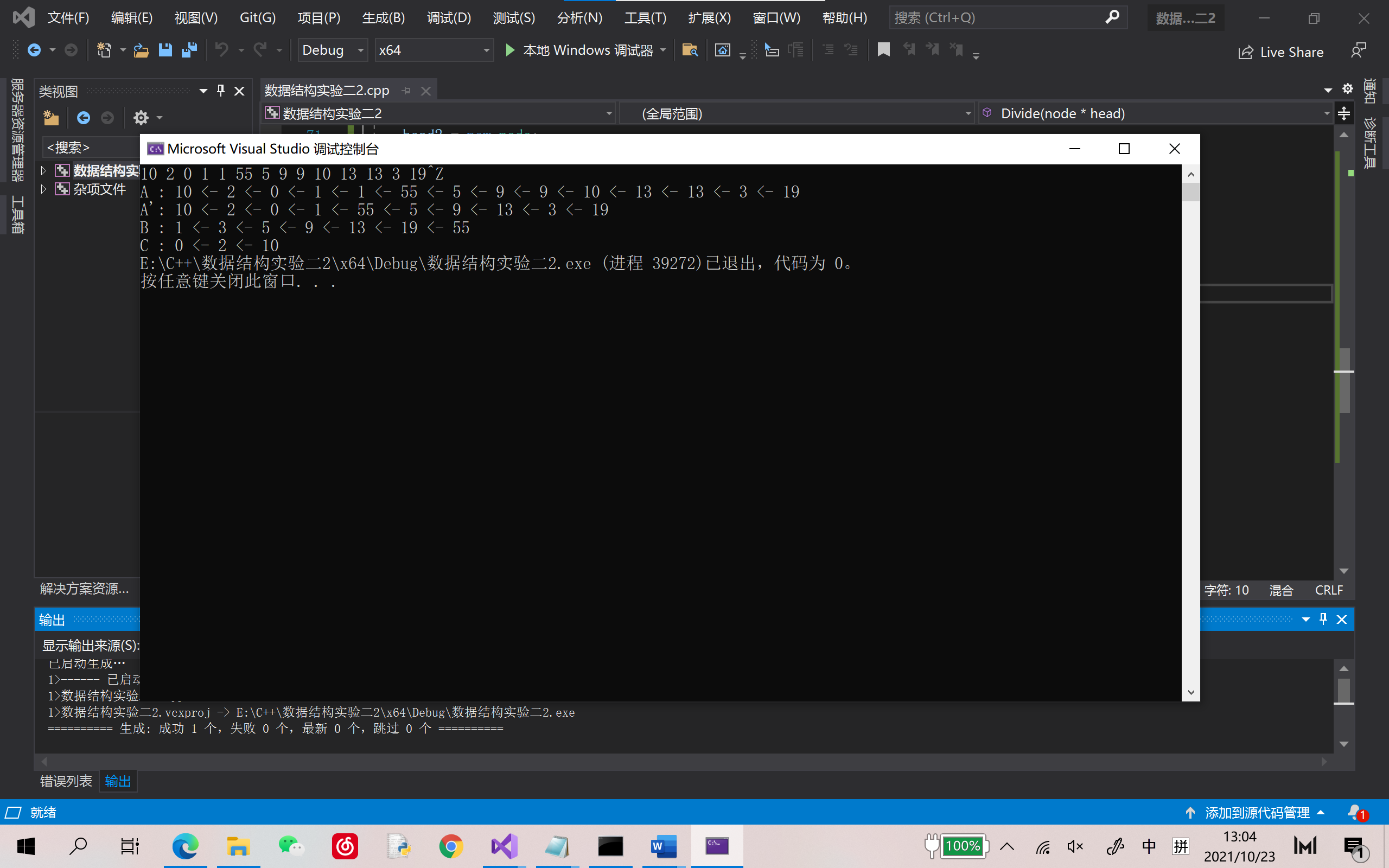
cout << endl << "C : ";

Output(head3->next);

return;

}

1. 运行结果



1. 实验分析
2. Insert函数
3. 空间复杂度
4. 时间复杂度
5. Delete函数

(1)空间复杂度

(2)时间复杂度

1. Divide函数

(1)空间复杂度

(2)时间复杂度