**单向链表实验报告**

1. 函数create，建立一个动态链表，各节点数据由键盘输入。
2. 函数print，输出前面操作后链表中的所有结点。
3. 函数del，删除指定结点。
4. 函数insert，插入一个结点；
5. 主函数，实现键盘输入，以及以上四个函数的功能。

**1建立结构体：**

struct student

{

int num;

int score;

struct student\* next;

};

内置struct 指向下一个结点的地址

**2.create函数：**

**/\* 建立链表，先给头文件一个空地址，**

**指针p1,p2分配空间，输入首个学生地址**

**接着按照输入的链表长度n依次输入个个**

**节点，n == 1时头指针指向p1，p2指向**

**p1新引入的地址，结束的时候p2->next指向空\*/**

student \*create() //创建链表

{

student \*head,\*p1,\*p2;

int n = 0;

head = NULL;

p1 = p2 = new student ;

cout<<"输入第1个学生学号和成绩:";

cin>>p1->num>>p1->score;//输入头结点

cout<<"\n";

while(p1->num!=0)

{

n+=1;

if(n == 1) head = p1;

else p2->next = p1;

p2 = p1;

p1 = new student;

cout<<"输入第"<<n+1<<"个学生学号和成绩:" ;

cin>>p1->num>>p1->score;

cout<<"\n";

}

p2->next = NULL;

return(head);

}

**3.函数print的实现：**

**/\*插入一个节点，按照学号排列\*/**

student\* insertstu(student \*head,student\* stud) //插入一个节点

{

int n=0;

student \*p0,\*p1,\*p2;

p1 = head; p0 =stud;

if(head == NULL) //如果链表为空，插入的p0就是head结点

head = p0, p0->next = NULL;

else //一般情况

{

while((p0->num>p1->num) && (p1->next != NULL))

p2 = p1 , p1 = p1->next;

if(p0->num<= p1->num)

{

if(p1 == head)

{

p0->next = head;

head = p0;

}

else

{

p2->next = p0;

p0->next = p1;

}

}

else

{

p1->next = p0;

p0->next = NULL;

}

}

n +=1;

return (head);

}

**4.函数insert的实现:**

**/\*插入一个节点，按照学号排列\*/**

student\* insertstu(student \*head,student\* stud) //插入一个节点

{

int n=0;

student \*p0,\*p1,\*p2;

p1 = head; p0 =stud;

if(head == NULL) //如果链表为空，插入的p0就是head结点

head = p0, p0->next = NULL;

else //一般情况

{

while((p0->num>p1->num) && (p1->next != NULL))

p2 = p1 , p1 = p1->next;

if(p0->num<= p1->num)

{

if(p1 == head)

{

p0->next = head;

head = p0;

}

else

{

p2->next = p0;

p0->next = p1;

}

}

else

{

p1->next = p0;

p0->next = NULL;

}

}

n +=1;

return (head);

}

5.主函数：

**/\*利用switch来实现比较良好的人机交互，达到用户输入什么实现什么的功能\*/**

int main()

{

int n;

student \*insertstu(student\*,student\*);

student\* del(student\* ,int );

student \*create();

void print(student \*head);

student\* head,\*stu;

int i=1,score,num;

while(i)

{

cout<<"输入想要执行的功能\n";

printf("1--建立新的链表\n");

printf("2--添加元素\n");

printf("3--删除元素\n");

printf("4--输出当前表中的元素\n");

printf("0--退出\n");

scanf("%d",&i);

switch(i)

{

case 0:

return 0;

break;

case 1:

head = create();

break;

case 2:

cout<<"请输入插入的学生学号与成绩\n";

stu = new stu;

cin>>stu->num>>stu->score;

insertstu(head,stu);

break;

case 3:

cout<<"请输入要删除的学生学号\n";

cin>>num;

del(head,num);

break;

case 4:

print(head);

break;

default:

printf("ERROR\n");

}

}

}

**思考与反思：**

在写部分函数的时候遇到了一定的难度，尤其是在删除的过程中发现以模板中的代码不能实现删除头节点的数，现在还是没有实现（脸黑）2019/3/5。

大早上的被迫发现的bug真的绝望，最后在主函数的case2中加入一行stu = new student;解决了第二次添加节点输出会无限循环的错误（在同一个地址里疯狂指向同样的地址。。。），所以在每次执行插入新结点操作时都要给stu一个新的地址，不然无限循环。。。2019/3/7