《数据结构与算法》实验报告

实验二：链表

**教 师：潘晔**

**学 生：梁书恺**

**学 号：2022040906023**

**时 间：10.9**

**地 点：科B119**

**一、ex1-1**

1. **问题分析：**
2. **题意理解**

从键盘输入学生信息，输入学号查询该学生，如果存在该学生，则删除，如果不存在，则在表尾添加。

1. **数据结构设计**

typedef struct Student

{

    long long id;

    char sex;

    char name[20];

    float score;

    struct Student \*next;

} Student;

1. **关键算法思路**

输入：链表头指针，学号

输出：该学号节点前驱的指针

指针p等于头指针；

当指针指向对象存在后继

如果指针指向对象后继的id匹配；

返回p；

指针在链表后移；

1. **健壮性设计**

通过调用统一出错处理函数ErrorHandler打印出错信息并跳出程序。

1. **性能分析**

时间复杂度：O(n);

空间复杂度：O(n);

1. **测试数据与运行结果截图：**

姓名: SkylerLiang

性别: M

学号：2022040906023

成绩: 100.00

姓名: Tiansuo

性别: M

学号：1145141919810

成绩: 88.80

姓名: FakeStudent

性别: F

学号：2022040906001

成绩: 666.00

请输入要查询的学号：2022040906023

找到了该学生

删除前：

姓名: SkylerLiang

性别: M

学号：2022040906023

成绩: 100.00

姓名: Tiansuo

性别: M

学号：1145141919810

成绩: 88.80

姓名: FakeStudent

性别: F

学号：2022040906001

成绩: 666.00

删除后：

姓名: Tiansuo

性别: M

学号：1145141919810

成绩: 88.80

姓名: FakeStudent

性别: F

学号：2022040906001

成绩: 666.00

姓名: SkylerLiang

性别: M

学号：2022040906023

成绩: 88.90

姓名: Tiansuo

性别: M

学号：1145141919810

成绩: 88.00

请输入要查询的学号：2022040906001

没有找到该学生

添加前：

姓名: SkylerLiang

性别: M

学号：2022040906023

成绩: 88.90

姓名: Tiansuo

性别: M

学号：1145141919810

成绩: 88.00

添加后：

姓名: SkylerLiang

性别: M

学号：2022040906023

成绩: 88.90

姓名: Tiansuo

性别: M

学号：1145141919810

成绩: 88.00

姓名: AddStudent

性别: M

学号：2022040906001

成绩: 0.00

1. **上机时遇到的问题**
2. **问题现象：** 学号输入异常 **原因：**学号范围过小，造成溢出； **解决办法：**使用long long作为id的类型
3. **问题现象：** 输入性别时直接跳过 **原因：**读到上一次输入的回车，作为字符存入性别中； **解决办法：**使用getchar()函数读取上次回车。
4. **程序代码**

ex1.c

ex1.h

**二、ex1-2**

1. **问题分析：**
2. **题意理解**

将两个递增有序的顺序表合并成一个递增有序的顺序表，合并后没有重复元素。

1. **数据结构设计**

typedef struct

{

    int length;

    int data[20];

} SeqL;

1. **关键算法思路**

输入：三个顺序表地址

输出：无

下标i，j，k

当A,B的下标都不超过其length

将A,B下标所指的数的较小者赋入C内，若一样，则赋一个值

当A下标小于其length

将剩余值都赋入C中

当B下标小于其length

将剩余值都赋入C中

1. **健壮性设计**

在处理前判断需要合并的顺序表是否递增有序，若不是，则退出。

1. **性能分析**

时间复杂度：O(n);

空间复杂度：O(n);

1. **测试数据与运行结果截图：**

A: 1 3 5 7 9

B: 2 4 6 8 10

C: 1 2 3 4 5 6 7 8 9 10

A: 1 3 5 7 9

B: 1 4 5 8 9

C: 1 3 4 5 7 8 9

A: 1 3 5 7 9

B: 1 4 8 5 9

顺序表不是递增有序的

1. **上机时遇到的问题**
2. **问题现象：** 两个被合并数组数据较多时出现问题 **原因：**顺序表data数组大小较小； **解决办法：**增大data数组大小
3. **程序代码**

ex2.c

ex2.h

**三、ex1-3**

1. **问题分析：**
2. **题意理解**

将两个递增有序的单链表合并成一个递增有序的顺序表，合并后没有重复元素。

1. **数据结构设计**

typedef struct Node

{

    int data;

    struct Node \*next;

} Node;

1. **关键算法思路**

Node \*Merge(Node \*A, Node \*B)

{

Node \*p = A->next;Node \*q = B->next;

Node \*r;Node \*s;Node \*t;

Node \*first = (Node \*)malloc(sizeof(Node));

if (first == NULL)

ErrorHandler(1);

first->next = NULL;

while (p != NULL && q != NULL)

{

if (p->data < q->data)

{

r = p;p = p->next;

r->next = first->next;

first->next = r;

}

else if (p->data > q->data)

{

r = q;q = q->next;

r->next = first->next;

first->next = r;

}

else

{

r = p;p = p->next;

s = q;q = q->next;

r->next = first->next;first->next = r;

s->next = first->next;first->next = s;

}

}

while (p != NULL)

{

r = p;p = p->next;

r->next = first->next;

first->next = r;

}

while (q != NULL)

{

r = q;q = q->next;

r->next = first->next;

first->next = r;

}

t = first;first = first->next;

free(t);

Delete(first);

return first;

}

1. **健壮性设计**

使用统一出错处理，可以处理内存分配出错和链表非递增的情况

1. **性能分析**

时间复杂度：O(n);

空间复杂度：O(n);

1. **测试数据与运行结果截图：**

A: 1 3 5 7 9

B: 2 4 6 8 10

合并AB后A: 9 8 7 6 5 4 3 2 1

C: 1 4 5 8 9

D: 2 4 6 8 10

合并CD后C: 9 8 6 5 4 2 1

E: 1 3 2 4 5

链表非递增

1. **上机时遇到的问题**
2. **问题现象：** 链表合并出现错误 **原因：**被合并链表非递增； **解决办法：**增加函数判断链表是否递增
3. **程序代码**

ex3.c

ex3.h

**四、ex1-4**

1. **问题分析：**
2. **题意理解**

约瑟夫环问题，给定人数n和报数m，程序输出每次离席人的序号。

1. **数据结构设计**

typedef struct Node

{

    int data;

    struct Node \*next;

}Node;

1. **关键算法思路**

输入：链表指针first，报数m

输出：无

下标i，循环指针p，辅助指针q

指针指向第一个数据节点；

当指针p指向节点的next不指向节点本身

i自增；

如果i等于m-1

q指向p的后继；

p指向节点的的next设为q的后继；

打印q所指节点的数据；

释放q；

i=0；

打印p指向节点的数据；

释放p；

1. **健壮性设计**

判断输入的数字是否不大于零，若是，则跳出程序。

1. **性能分析**

时间复杂度：O(n\*m);

空间复杂度：O(n);

1. **测试数据与运行结果截图：**

请输入人数n和报数m:10,3

3 6 9 2 7 1 8 5 10 4

请输入人数n和报数m:20,666

6 7 5 10 20 9 18 2 13 1 14 12 16 17 15 19 4 3 11 8

请输入人数n和报数m:100,-7

输入有误

1. **上机时遇到的问题**
2. **问题现象：** 在while循环中逻辑有误 **原因：**在while循环中不应该使用for循环； **解决办法：**在while循环里直接自增
3. **问题现象：** 求解约瑟夫环时总是错位 **原因：**第一个节点为头节点； **解决办法：**将指针指向第一个数据节点再开始求解。
4. **程序代码**

ex4.c

ex4.h

**小结**

这次上机实验比较复杂，链表在使用时比较容易出错，在编写函数之前，应先根据功能需求，以及函数内是否需要修改参数的实值，来判断函数是否应该传入指针变量。在使用指针指向节点的时候，要注意指针的使用，是对结点的next进行操作还是对指针进行操作。