

学号	2021144301	姓名	王大殿	实验日期	
实验名称	结构体与单链表				
实验目的： <div>1. 掌握结构体类型的说明、结构体变量(数组)的定义及初始化方法； 2. 掌握结构体变量成员的引用；。</div>					
实验任务、步骤与结果： <div>任务 1：求圆的周长与面积 题目描述</div> <div>1. 学生成绩录入与输出</div> <div>设计结构体并声明变量，存储学生学号，姓名，语，数，外三门课程成绩。 函数 DataInput() DataOutput() 分别完成数据输与输出</div> <div>【输入要求】 N+1 行： 第 1 行，输入一个正整数 N（N <= 100），表示学生人数。 第 2 至接着输入 N+1 行，第 1~N 个学生信息，每行格式如下： 学号 姓名 分数 1 分数 2 分数 3 分数是一个非负整数，且小于等于 100； 姓名为一个连续的字符串，中间没有空格，长度不超过 20。</div> <div>【输出要求】 N+1 行： 第 1 行，表头，第 2~N+1 行，学生成绩信息。 成绩精确到小数点后面一位 学号 姓名 分数之间的间距为一个 Tab 代码： #include<stdio.h> #include<malloc.h> #include<string.h></div>					

```

typedef int Status;

//学生信息内容
typedef struct{
    char stu_ID[10];
    char name[20];
    float score[3];
}student;

typedef student ElemType;

//链表结构
typedef struct LNode{
    ElemType data;
    struct LNode *next;
}LNode, *LinkList;

//链表初始化
Status InitList(LinkList *L);

//显示学生信息
void Output(ElemType *e);

//获取学生信息
Status GetElem(LinkList L, int i, ElemType *e);

//在指定位置插入某个学生信息
Status ListInsert(LinkList L, int i, ElemType e);

//根据名字进行查找
Status Search(LNode L, char str[], LinkList *p);

//删除指定位置的学生信息
Status ListDelete(LinkList L, int i);

//输入学生信息
void Input(ElemType *e);

int main(){
    LinkList L;
    int n, choose; //学生个数以及菜单序号
    ElemType a, b; //用于暂存学生信息
    printf("\n1.构造链表\n");
    printf("2.输入学生信息\n");

```

```

printf("3.显示学生表信息\n");
printf("9.退出\n\n");
while(1){
    printf("请选择: ");
    scanf("%d",&choose);
    if(choose==9) break;
    switch(choose){
        case 1: if(InitList(&L)){
                    printf("成功建立链表\n");
                }else{
                    printf("失败");
                }
                break;
        case 2: printf("请输入要录入的学生人数: ");
                scanf("%d",&n);
                for(int i=1; i<=n; i++){
                    printf("第%d 个学生: \n",i);
                    Input(&a);
                    ListInsert(L, i, a);
                }
                break;
        case 3: for(int i=1; i<=n; i++){
                    GetElem(L,i,&b);
                    Output(&b);
                }
                break;
    }
}

Status InitList(LinkList *L){
    (*L)=(LinkList)malloc(sizeof(LNode));
    (*L)->next=NULL;
    return 1;
}

void Output(ElemType *e){
    printf("学号: %s\t 姓名: %s\t 成绩 1: %.1f\t 成绩 2: %.1f\t 成绩 3: %.1f\n",
        e->stu_ID,e->name,e->score[0],e->score[1],e->score[2]);
}

Status GetElem(LinkList L, int i, ElemType *e){
    LinkList P;
    P=L->next;
    int j=1;
    while (P&& j<i){

```

```

        //P 不为空, j 小于 i
        P=P->next;
        j++;
    }
    if(!P || j>i){
        //如果 P 为空, 或 j 大于 i
        return 0;
    }
    *e=P->data;
    return 1;
}
Status ListInsert(LinkList L, int i, ElemType e){
    LinkList P, S;
    P=L;
    int j=0;
    while(P && j<i-1){
        //j 小于 i-1, 空出 i 的位置
        P=P->next;
        j++;
    }
    if(!P || j>i-1){
        return 0;
    }
    S=(LinkList)malloc(sizeof(LNode));
    S->data=e; //把 e 的值赋给新节点的数据部分
    S->next=P->next; //把第 i-1 个节点的下一个节点位置赋给新建节点的 next
    P->next=S; //第 i-1 的 next 指向新建节点
    return 1;
}
void Input(ElemType *e){
    printf("学号: "); scanf("%s",e->stu_ID);
    printf("姓名: "); scanf("%s",e->name);
    printf("成绩 1: "); scanf("%f",&e->score[0]); //这里一定要用取地址符, 原因未知?
    printf("成绩 2: "); scanf("%f",&e->score[1]);
    printf("成绩 3: "); scanf("%f",&e->score[2]);
    printf("输入完成\n");
}

```

运行结果

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验10$ ./学生信息管理
```

```
1.构造链表
2.输入学生信息
3.显示学生表信息
9.退出
```

```
请选择: 1
```

```
成功建立链表
```

```
请选择: 2
```

```
请输入要录入的学生人数: 2
```

```
第1个学生:
```

```
学号: 2001
```

```
姓名: wdd
```

```
成绩1: 10
```

```
成绩2: 29
```

```
成绩3: 12
```

```
输入完成
```

```
第2个学生:
```

```
学号: 2002
```

```
姓名: wer
```

```
成绩1: 29
```

```
成绩2: 34
```

```
成绩3: 12
```

```
输入完成
```

```
请选择: 3
```

```
学号: 2001      姓名: wdd      成绩1: 10.0      成绩2: 29.0      成绩3: 12.0
```

```
学号: 2002      姓名: wer      成绩1: 29.0      成绩2: 34.0      成绩3: 12.0
```

```
请选择: 9
```

3 约瑟夫（链表）

用链表实现约瑟夫问题：13 个人围成一圈，从第 1 个人开始顺序报号 1，2，3。凡报到 3 者退出圈子。找出最后留在圈子中的人原来的序号。

代码：

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
typedef struct node {
```

```
    int number;
```

```
    struct node * next;
```

```
}person;
```

```
person * initLink(int n) {
```

```
    int i = 0;
```

```
    person * head = NULL, *cyclic = NULL;
```

```
    head = (person*)malloc(sizeof(person));
```

```
    head->number = 1;
```

```
    head->next = NULL;
```

```
    cyclic = head;
```

```
    for (i = 2; i <= n; i++) {
```

```
        person * body = (person*)malloc(sizeof(person));
```

```
        body->number = i;
```

```
        body->next = NULL;
```

```
        cyclic->next = body;
```

```
        cyclic = cyclic->next;
```

```
    }
```

```
    cyclic->next = head;//首尾相连
```

```

        return head;
    }

    void findAndKillK(person * head, int k, int m) {
        person * p = NULL;
        person * tail = head;
        //找到链表第一个结点的上一个结点，为删除操作做准备
        while (tail->next != head) {
            tail = tail->next;
        }
        p = head;
        //找到编号为 k 的人
        while (p->number != k) {
            tail = p;
            p = p->next;
        }
        //从编号为 k 的人开始，只有符合 p->next==p 时，说明链表中除了 p 结点，所有
        //编号都出列了，
        while (p->next != p) {
            int i = 0;
            //找到从 p 报数 1 开始，报 m 的人，并且还要知道数 m-1de 人的位置 tail，方
            //便做删除操作。
            for (i = 1; i < m; i++) {
                tail = p;
                p = p->next;
            }
            tail->next = p->next; //从链表上将 p 结点摘下来
            printf("出列人的编号为:%d\n", p->number);
            free(p);
            p = tail->next; //继续使用 p 指针指向出列编号的下一个编号，游戏继续
        }
        printf("出列人的编号为:%d\n", p->number);
        free(p);
    }

    int main() {
        int n = 0, k = 0, m = 0;
        person * head = NULL;
        printf("输入圆桌上的人数:");
        scanf("%d", &n);
        head = initLink(n);
        printf("从第几个人开始报数(k>1 且 k<=%d): ", n);
        scanf("%d", &k);
        printf("数到几的人出列: ");
    }

```

```
scanf("%d", &m);  
findAndKillK(head, k, m);  
return 0;  
}  
运行结果:
```

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验10$ ./约瑟夫_链表  
输入圆桌上的人数:10  
从第几个人开始报数(k>1且k<10): 3  
数到几的人出列: 5  
出列人的编号为:7  
出列人的编号为:2  
出列人的编号为:8  
出列人的编号为:4  
出列人的编号为:1  
出列人的编号为:10  
出列人的编号为:3  
出列人的编号为:6  
出列人的编号为:9  
出列人的编号为:5
```

发现问题与分析:

备注:

说明:

- 1、各项记录应完备详尽。
- 2、各栏目如果长度不够可以自行调整大小，但应注意排版的美观。
- 3、可将实现过程的代码、运行效果等内容以贴图方式放在相应栏目。