

桂林电子科技大学 2022-2023 学年 第 1 学期

程序设计与问题求解 实验报告

课号: 2212749

任课老师: 甘才军

实验名称 实验六-链表程序设计

院 系	计算机与信息安全学院	专业	网络空间安全
-----	------------	----	--------

学 号	2200350204	姓名	李禹佳
-----	------------	----	-----

实验日期	2022 年 12 月 3 日
------	-----------------

1. 实验目的

1. 掌握结构化数据的编程使用;
2. 掌握指针与内存地址的关系;
3. 掌握通过指针动态申请和释放内存的编程方法;
4. 掌握单向链表的基本操作。

6. 实验内容与步骤

实验任务: 详见 educoder 平台普通作业“实验六-链表程序设计”

程序的编号: 严格遵循统一规范, 详见本课号的 QQ 群。

请在下面贴出各个程序源码截图和程序运行画面截图(至少包含 3 组测试用例, 确保测试用例覆盖所有典型场景)。程序运行结果画面要包含你的学号和姓名信息。

1. 结构体相关编程:改正下列程序错误,使之完成规定功能

程序实现输出图书的名字和单价, 错误代码如下:

```
#include struct book {  
    float price;//价格  
    char name[10];//名字 }  
int main(void) {  
    struct book myBook;  
    myBook={5.6,"the world is flat"};  
    printf("book name=%s,book price=%f",myBook.name,myBook.price);  
    return 0; }
```

改后正确代码如下

```
liyujiazezuoyeaaa.cpp x
1  #include <stdio.h>
2  struct book
3  {
4  float price;//价格
5  char name[100];//名字
6  };
7  int main()
8  {
9  struct book myBook={5.6,"the world is flat"};
10 printf("book name=%s,book price=%f\n",myBook.name,myBook.price);
11 printf("2200350204 李禹佳");
12 return 0;
13 }
14
```

运行结果如下:

```
book name=the world is flat,book price=5.600000
2200350204 李禹佳
-----
Process exited after 0.04796 seconds with return value 0
请按任意键继续. . .
```

2. 输入若干学生的信息（学号、姓名、成绩），当输入学号为 0 时结束，用单向链表组织这些学生信息后，再按序输出。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct stud_node {
    int num;
    char name[20];
    int score;
    struct stud_node *next;
};
int main() {
    struct stud_node *head, *tail, *p;
    int num, score;
    char name[20];
    int size = sizeof(struct stud_node);
    head = tail = NULL;
    printf("input num,name and score:\n");
    scanf("%d", &num);
    while (num != 0) {
        p = (struct stud_node*)malloc(size);
        scanf("%s %d", name, &score);
        p->num = num;
        strcpy(p->name, name);
        p->score = score;
        p->next = NULL;
        if(head==NULL){

```

```

18 while (num != 0) {
19     p = (struct stud_node*)malloc(size);
20     scanf("%s %d", name, &score);
21     p->num = num;
22     strcpy(p->name, name);
23     p->score = score;
24     p->next = NULL;
25     if(head==NULL){
26         head=tail=p;
27     }else{
28         tail->next=p;
29         tail=p;
30     }
31     scanf(" %d", &num);
32 }
33 for (p = head; p->next != NULL; p = p->next)
34     printf(" %d %s %d\n", p->num, p->name, p->score);
35 return 0;
36 }

```

3. (1) 建立一个单链表 21 3 15 27 11 18，并输出该链表；
(2) 输入序号 n，查找序号为 n 的结点，并输出；
(3) 输入值 x，查找值为 x 的结点，并输出；
(4) 插入结点: 输入序号 n 和值 x。在序号为 n 的结点后插入 x，并输出该链表；
(5) 删除结点: 输入序号 n，删除序号为 n 的结点，并输出该链表。

下面为代码(附注释)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #define LEN sizeof(struct Nmb)
4  typedef struct Nmb { //创建单向链表, 包含值number和xuhao
5      int number;
6      int xuhao;
7      struct Nmb *next; //指向下一个节点的头部
8  } Num;
9
10 Num *createLink(Num *head);
11 void printLink(Num *head);
12 void FindNode(Num *head, int n);
13 void FindNum(Num *head, int x);
14 Num *insertNode(Num *head, int n, int x);
15 Num *DeleteNobe(Num *head, int n);
16
17 int main() {
18     int n, x;
19     Num *head = NULL;
20     head = createLink(head); // (1) 逐个输出节点的值, 输出链表(从头部开始输出结点的值)
21
22     printLink(head); // 打印换行
23     printf("\n");
24
25     scanf("%d", &n); // (2) 输入整数n, 输出序号为n的节点值
26     FindNode(head, n); // 打印换行
27     printf("\n");
28
29     scanf("%d", &x); // (3) 输入整数x, 查找第一个值为x的节点, 并输出其序号和值
30     FindNum(head, x); // 打印换行
31     printf("\n");
32
33     scanf("%d%d", &n, &x); // (4) 插入序号为n值为x的节点
34     head = insertNode(head, n, x);
35     printLink(head); // 打印换行
36     printf("\n");
37
38     scanf("%d", &n); // (5) 输入序号n, 删除序号为n的节点
39     head = DeleteNobe(head, n);
40     printLink(head);
41
42     return 0;
43 }
```

```

44 Num *createlink(Num *head) {//(1)
45     int number, i = 1;
46     Num *tail = head; //结点的尾部指向头部
47     Num *p; //创建节点
48     scanf("%d", &number); //第一个节点的值
49     while (number != 0) {
50         p = (Num*)malloc(sizeof(Num));
51         p->number = number; //节点中Number的值为输入的number值
52         p->xuhao = i; //结点的序号循环递增
53         i++;
54         p->next = NULL; //头个节点指针为空,其余分别指向下一个节点
55         if (head == NULL) {
56             head = p;
57             tail = p;
58         } else {
59             tail->next = p;
60             tail = p;
61         }
62         scanf("%d", &number); //循环输入每一个结点的值 number
63     }
64     return head;
65 }

```

```

67 void printLink(Num *head) { //打印输入的节点值
68     Num *p;
69     for (p = head; p != NULL; p = p->next) {
70         if (p->xuhao == 1) //第一个节点值前面没有空格
71             printf("%d", p->number);
72         else
73             printf(" %d", p->number);
74     }
75 }
76
77 void FindNode(Num *head, int n) {
78     Num *p;
79     for (p = head; p != NULL; p = p->next) {
80         if (p->xuhao == n) {
81             printf("%d", p->number); //查找序号
82         }
83     }
84 }
85

```

```

86 void FindNum(Num *head, int x) {
87     Num *p;
88     for (p = head; p != NULL; p = p->next) {
89         if (p->number == x) {
90             printf("%d %d", p->xuhao, p->number); //查找值
91         }
92     }
93 }
94
95 Num *insertNode(Num *head, int n, int x) {
96     Num *p, *t;
97     for (p = head; p != NULL; p = p->next) {
98         if (p->xuhao == n) {
99             t = (Num*)malloc(sizeof(Num));
100             t->number = x;
101             t->xuhao = n + 1; //开辟新节点t
102             t->next = p->next;
103             p->next = t;
104         }
105     }
106     for (p = t->next; p != NULL; p = p->next) {
107         p->xuhao = p->xuhao + 1;
108     }

```

```

106     for (p = t->next; p != NULL; p = p->next) {
107         p->xuhao = p->xuhao + 1;
108     }
109     return head;
110 }
111
112 Num *DeleteNobe(Num *head, int n) {
113     Num *p;
114     int flag = n - 1;
115     for (p = head; p != NULL; p = p->next) {
116         if (p->xuhao == flag) {
117             Num *t = (Num*)malloc(sizeof(Num));
118             t = p->next;
119             p->next = t->next;
120             free(t); //释放空间
121         }
122     }
123     int i = n;
124     while (p != NULL) {
125         p->xuhao = i;
126         i++;
127         p = p->next;

```

```

Num *DeleteNobe(Num *head, int n) {
    Num *p;
    int flag = n - 1;
    for (p = head; p != NULL; p = p->next) {
        if (p->xuhao == flag) {
            Num *t = (Num*)malloc(sizeof(Num));
            t = p->next;
            p->next = t->next;
            free(t); //释放空间
        }
    }
    int i = n;
    while (p != NULL) {
        p->xuhao = i;
        i++;
        p = p->next;
    }
    return head;
}

```

Free(*head);

下方为运行结果

```

21 3 15 27 11 18 0
21 3 15 27 11 18
3
15
15
3 15
3 33
21 3 15 33 27 11 18
4
21 3 15 27 11 182200350204 李禹佳

```