

试讲课

讲师介绍：石璞东，西安建筑科技大学信息与控制工程学院硕士

考试科目：883 《C程序设计》（第五版）谭浩强，清华大学出版社

参考资料：《C程序设计（第五版）》谭浩强

目录

- 课程时长：2.5小时
 - 第一章 程序设计和 C 语言
 - 第二章 算法—程序的灵魂
- 课程时长：6小时
 - 第三章 最简单的 C 程序设计—顺序程序设计
 - 第四章 选择结构程序设计
 - 第五章 循环结构程序设计
- 课程时长：6小时
 - 第六章 利用数组批量处理数据
 - 第七章 用函数实现模块化程序设计
 - 第八章 善于利用指针
- 课程时长：6小时
 - 第九章 用户自己建立数据类型
 - 第十章 对文件的输入输出
- 课程时长：2小时
 - 试卷解析

案例展示

1. 求阶乘

```
1  #include<stdio.h>
2  int main(){
3      int num,temp = 1;
4      printf("请输入num数值: ");
5      scanf("%d",&num);
6      if(num>=0){
7          for(int i = 1;i <= num;i++){
8              temp *= i;
9          }
10         printf("%d的阶乘值为: %d\n",num,temp);
```

```

11     }else{
12         printf("请重新输入num数值! ");
13     }
14     return 0;
15 }

```

效果展示：

```

#include<stdio.h>
int main(){
    int num,temp = 1;
    printf("请输入num数值: ");
    scanf("%d",&num);
    if(num>=0){
        for(int i = 1;i <= num;i++){
            temp *= i;
        }
        printf("%d的阶乘值为: %d\n",num,temp);
    }else{
        printf("请重新输入num数值! ");
    }
    return 0;
}

```

```

请输入num数值: 5
5的阶乘值为: 120
Program ended with exit code: 0

```

2. 判断闰年

```

1  #include <stdio.h>
2  int main(){
3      printf("2000年~2500年中的闰年年份包括: \n");
4      for(int a = 2000;a <= 2500;a++){
5          if((a%4==0&&a%100!=0) || (a%400==0)){
6              printf("%d\n",a);
7          }
8      }
9      return 0;
10 }

```

效果实战：

```

59
60 #include <stdio.h>
61 int main(){
62     printf("2000年~2500年中的闰年份包括: \n");
63     for(int a = 2000; a <= 2500; a++){
64         if((a%4==0&&a%100!=0)|| (a%400==0)){
65             printf("%d\n", a);
66         }
67     }
68     return 0;
69 }
70
71
72
73
74 /*

```

2000年~2500年中的闰年份包括:

2000
 2004
 2008
 2012
 2016
 2020
 2024
 2028
 2032
 2036
 2040
 2044
 2048
 2052
 2056
 2060
 2064
 2068
 2072
 2076
 2080
 2084
 2088
 2092
 2096
 2104





Filter

All Output

3. 链表实战

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  #include<stack>
5
6  using namespace std;
7  struct student{
8      int num;//学号
9      char name[20];//姓名
10     int age;//年龄
11     struct student *next;//指向下一个节点的指针
12 };
13
14 struct student *createList();//建立学生信息
15 void displayList(struct student *head);//按照建立顺序输出学生信息
16 struct student *reverseList(struct student *head);//与建立顺序相反, 逆序顺序输出学生信息
17 struct student *deleteNodes(struct student *head);//删除某个学生信息
18 struct student *insertNodes(struct student *head);//插入某个学生的信息

```

```

19 void search_student_info(struct student *head); //查找对应学生信息
20 void modify_student_info(struct student *head); //修改学生信息
21 struct student *destroyList(struct student *head); //清空整个链表
22 struct student *head = NULL;
23
24 int main(){
25     int select; //根据对应不同select的值,调用不同的函数
26     do{
27         printf("请输入select的值:");
28         scanf("%d",&select);
29         switch(select){
30             case 1:
31                 head = createList();
32                 break;
33             case 2:
34                 displayList(head);
35                 break;
36             case 3:
37                 head = deleteNodes(head);
38                 break;
39             case 4:
40                 head = insertNodes(head);
41                 break;
42             case 5:
43                 search_student_info(head);
44                 break;
45             case 6:
46                 modify_student_info(head);
47                 break;
48             case 7:
49                 head = reverseList(head);
50             //     displayList(head);
51                 break;
52             case 8:
53                 head = destroyList(head);
54                 break;
55
56         }
57     }while(select!=0);
58     return 0;
59 }
60 //1. 建立学生信息
61 struct student *createList(){
62     struct student *head; //头节点
63     struct student *p1; //开辟新节点
64     struct student *p2; //与p1连接
65     int num1;
66     char name1[20];
67     int age1;

```

```

68     head = NULL;
69     int count = 1;
70
71     printf("请输入第1个学生的学号、姓名、年龄(用空格分隔):");
72     scanf("%d%s%d", &num1, &name1, &age1);
73     while(age1>0){
74         p1 = (struct student*)malloc(sizeof(struct student));
75         p1->num = num1;
76         strcpy(p1->name, name1);
77         p1->age = age1;
78         p1->next = NULL;
79         if(head == NULL){
80             head = p1;
81         }else{
82             p2->next = p1;
83         }
84         p2 = p1;
85         printf("请输入第%d个学生的学号、姓名、年龄(用空格分隔):", ++count);
86         scanf("%d%s%d", &num1, &name1, &age1);
87     }
88     return head;
89 }
90
91 //2. 与建立顺序相同输出学生信息
92 void displayList(struct student *head){
93     struct student *p;
94     int n = 0;
95     if(head!=NULL){
96         printf("顺序输出链表中学生信息如下:\n");
97         for(p=head;p!=NULL;p=p->next){
98             printf("学号:%-6d 姓名:%-20s 年龄:%-6.1d\n", p->num, p->name, p->age);
99             n++;
100         }
101         printf("学生总数:%d\n", n);
102     }else{
103         printf("空链表!\n");
104     }
105 }
106
107 //3. 根据学号删除对应学生信息
108 struct student *deleteNodes(struct student *head){
109
110     struct student *p1;
111     struct student *p2;
112
113     int num2;//要删除学生的学号
114     printf("请输入要删除学生的学号:");
115     scanf("%d", &num2);
116

```

```

117     if(head == NULL){
118         printf("链表为空\n");
119         return head;
120     }
121     p2 = head;
122
123     while(num2!=p2->num&& p2->next!=NULL){ //查找要删除的节点
124         p1 = p2;
125         p2 = p2->next;
126     }
127
128     if(num2 == p2->num){
129         if(p2 == head){ //要删除的是头节点
130             head = p2->next;
131         }else{ //其他节点
132             p1->next = p2->next;
133         }
134         free(p2);
135         printf("删除了学号为%d的学生信息!\n", num2);
136     }else{
137         printf("该生不存在!\n");
138     }
139     return head;
140 }
141
142 //4. 根据学号的大小插入某个学生的信息
143 struct student *insertNodes(struct student *head){
144     struct student *p; //待插入节点
145     struct student *p1; //待插入节点的前驱节点
146     struct student *p2; //待插入节点的后继节点
147     p2 = head;
148     p = (struct student *)malloc(sizeof(struct student));
149     printf("请输入要加入学生的学号、姓名、年龄:");
150     scanf("%d%s%d", &p->num, &p->name, &p->age);
151     if(head == NULL){ //若为空链表,则相当于创建一个新节点
152         head = p;
153         p->next = NULL;
154     }else{
155         while(p->num > p2->num&& p2->next!=NULL){ //查找待插入的位置
156             p1 = p2;
157             p2 = p2->next;
158         }
159         if(p->num < p2->num){ //头节点和中间任意节点的插入
160             if(p == head){ //头节点
161                 head = p;
162                 p->next = p1;
163             }else{ // 中间任意节点
164                 p1->next = p;
165                 p->next = p2;

```

```

166     }
167     }else{//尾节点的插入
168         p2->next = p;
169         p->next = NULL;
170     }
171 }
172     return head;
173 }
174
175 //5. 根据学号查找对应学生的其他信息
176 void search_student_info(struct student *head){
177     struct student *p;
178     int num;//要查找对应学生的学号信息
179     printf("请输入要查找学生的学号:");
180     scanf("%d",&num);
181     p = head;
182     //非空链表的情况下
183     if(head != NULL){
184         while(p->num!=num&& p->next!=NULL){
185             p=p->next;
186         }
187         //不满足while循环的第一个条件
188         if(p->num==num){
189             printf("你所查找的学号为%d的学生信息如下:\n",num);
190             printf("学号:%-6d 姓名:%-20s 年龄:%-6.1d\n",p->num,p->name,p->age);
191         }else{
192             //不满足while循环的最后一个条件
193             printf("没有找到学号为%d的学生信息,请确认学号是否正确!\n",num);
194         }
195     }else{//空链表的情况
196         printf("空链表!");
197     }
198 }
199
200 //6. 根据学号修改学生信息
201 void modify_student_info(struct student *head){
202     struct student *p;
203     int num;
204     char name[20];
205     int age;
206     printf("请输入您要修改学生的学号:");
207     scanf("%d",&num);
208     p = head;
209     if(head!=NULL){
210         while(p->num!=num&& p->next!=NULL){
211             p = p->next;
212         }
213         if(p->num==num){
214             printf("请输入您要更改后的信息:\n");

```

```

215     scanf("%d%s%d", &num, &name, &age);
216     p->num = num;
217     strcpy(p->name, name);
218     p->age = age;
219 }else{
220     printf("没有找到学号为%d的学生信息,请确认学号信息是否正确!\n");
221 }
222 }else{
223     printf("空链表\n");
224 }
225
226
227 }
228
229 //7. 与建立顺序相反输出学生信息
230 struct student *reverseList(struct student *head){
231     /*
232         1. 借助递归
233     */
234     /*
235     struct student *p;
236     p = head;
237     if(p!=NULL){
238         reverseList(p->next);
239         printf("学号:%-6d 姓名:%-20s 年龄:%-6.1d\n", p->num, p->name, p->age);
240     }
241     */
242
243     /*
244         2. 借助栈
245     */
246     struct student *p;
247     p = head;
248     stack<int> s;
249     while(p!=NULL){
250         s.push(p->num);
251         p = p->next;
252     }
253     while(!s.empty()){
254         p = head;
255         while(s.top()!=p->num){
256             p = p->next;
257         }
258         if(s.top()==p->num){
259             printf("学号:%-6d 姓名:%-20s 年龄:%-6.1d\n", p->num, p->name, p->age);
260         }
261         s.pop();
262     }
263     return head;

```



```

264
265  /*
266     3. 改变单链表指针指向
267     改变指针指向后，原链表也会被修改
268  */
269
270  /*
271  struct student *pre;
272  struct student *post;
273  struct student *p;
274  pre = NULL;
275  post = NULL;
276  while(head!=NULL){
277      post = head->next;
278      head->next = pre;
279      pre = head;
280      head = post;
281  }
282  return pre;
283  */
284
285  }
286
287  //8. 清空整个链表
288  struct student *destroyList(struct student *head){
289      struct student *p;
290      p = head;
291      if(p==NULL){
292          printf("空链表!\n");
293      }
294      while(p!=NULL&& p->next!=NULL){
295          p = p->next;
296          free(p);
297      }
298      printf("信息删除完毕!\n");
299      head = NULL;
300      return head;
301  }

```

效果展示：

```
C:\Users\TURBO\Desktop\数据结构\算法练习\链表练习.exe
请输入select的值:1 建立链表
请输入第1个学生的学号、姓名、年龄(用空格分隔):1 西安科技大学 18
请输入第2个学生的学号、姓名、年龄(用空格分隔):2 石璞东 18
请输入第3个学生的学号、姓名、年龄(用空格分隔):3 蕊希 18
请输入第4个学生的学号、姓名、年龄(用空格分隔):0 0
请输入select的值:3 删除学生信息
请输入要删除学生的学号:1
删除了学号为1的学生信息!
请输入select的值:2 顺序输出信息
顺序输出链表中学生信息如下:
学号:2 姓名:石璞东 年龄:18
学号:3 姓名:蕊希 年龄:18
学生总数:2
请输入select的值:4 插入学生信息
请输入要加入学生的学号、姓名、年龄:5 翠花 18
请输入select的值:4
请输入要加入学生的学号、姓名、年龄:4 宋思迪 18
请输入select的值:2
顺序输出链表中学生信息如下:
学号:2 姓名:石璞东 年龄:18
学号:3 姓名:蕊希 年龄:18
学号:4 姓名:宋思迪 年龄:18
学号:5 姓名:翠花 年龄:18
学生总数:4
请输入select的值:5 查找学生信息
请输入要查找学生的学号:3
你所查找的学号为3的学生信息如下:
学号:3 姓名:蕊希 年龄:18
请输入select的值:6 修改学生信息
请输入您要修改学生的学号:3
请输入您要更改后的信息:
3 蕊希 3
请输入select的值:2
顺序输出链表中学生信息如下:
学号:2 姓名:石璞东 年龄:18
学号:3 姓名:蕊希 年龄:3
学号:4 姓名:宋思迪 年龄:18
学号:5 姓名:翠花 年龄:18
学生总数:4
请输入select的值:7 逆序输出
学号:5 姓名:翠花 年龄:18
学号:4 姓名:宋思迪 年龄:18
学号:3 姓名:蕊希 年龄:3
学号:2 姓名:石璞东 年龄:18
请输入select的值:8 清空链表
信息删除完毕!
请输入select的值:2
空链表!
请输入select的值:
```

文档资料链接：

- Github: https://github.com/TURBO1002/C_program_language_FOR_graduate_student
- Gitee: https://gitee.com/turbodong/c_program_language_for_graduate_student