

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期： 2019.12.20**

**网络空间安全学院**

**目 录**

**[7 结构与联合实验 94](#_Toc27818360)**

[7.1 实验目的 94](#_Toc27818361)

[7.2 实验内容 94](#_Toc27818362)

[7.3 实验小结 113](#_Toc27818367)

**[参考文献 114](#_Toc27818368)**

**7 结构与联合实验**

**7.1实验目的**

1．通过实验，熟悉和掌握结构的说明和引用、结构的指针、结构数组、以及函数中使用结构的方法。

2．通过实验，掌握动态储存分配函数的用法，掌握自引用结构，单向链表的创建、遍历、结点的增删、查找等操作。

3．了解字段结构和联合的用法。

**7.2实验内容**

**7.2.1表达式求值的程序验证题**

设有说明：

char u[]="UVWXYZ";

char v[]="xyz";

struct T{

int x;

char c;

char \*t;

}a[]={{11,ˊAˊ,u},{100, ˊBˊ,v}},\*p=a;

请先自己计算下面表达式的值，然后通过编程计算来加以验证。(各表达式相互无关)

序号 表达式 计算值 验证值

1 (++p)->x 100 100

2 p++,p->c ‘B’ ‘B’

3 \*p++->t,\*p->t ‘x’ ‘x’

4 \*(++p)->t ‘x’ ‘x’

5 \*++p->t ‘V’ ‘V’

6 ++\*p->t ‘V’ ‘V’

**7.2.2源程序修改替换题**

给定一批整数，以0作为结束标志且不作为结点，将其建成一个先进先出的链表，先进先出链表的指头指针始终指向最先创建的结点（链头），先建结点指向后建结点，后建结点始终是尾结点。

（1） 源程序中存在什么样的错误（先观察执行结果）？对程序进行修改、调试，使之能够正确完成指定任务。

源程序如下：

1 #include "stdio.h"

2 #include "stdlib.h"

3 struct s\_list{

4 int data; /\* 数据域 \*/

5 struct s\_list \*next; /\* 指针域 \*/

6 } ;

7 void create\_list (struct s\_list \*headp,int \*p);

8 void main(void)

9 {

10 struct s\_list \*head=NULL,\*p;

11 int s[]={1,2,3,4,5,6,7,8,0}; /\* 0为结束标记 \*/

12 create\_list(head,s); /\* 创建新链表 \*/

13 p=head; /\*遍历指针p指向链头 \*/

14 while(p){

15 printf("%d\t",p->data); /\* 输出数据域的值 \*/

16 p=p->next; /\*遍历指针p指向下一结点 \*/

17 }

18 printf("\n");

19 }

20 void create\_list(struct s\_list \*headp,int \*p)

21 {

22 struct s\_list \* loc\_head=NULL,\*tail;

23 if(p[0]==0) /\* 相当于\*p==0 \*/

24 ;

25 else { /\* loc\_head指向动态分配的第一个结点 \*/

26 loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));

27 loc\_head->data=\*p++; /\* 对数据域赋值 \*/

28 tail=loc\_head; /\* tail指向第一个结点 \*/

29 while(\*p){ /\* tail所指结点的指针域指向动态创建的结点 \*/

30 tail->next=(struct s\_list \*)malloc(sizeof(struct s\_list));

31 tail=tail->next; /\* tail指向新创建的结点 \*/

32 tail->data=\*p++; /\* 向新创建的结点的数据域赋值 \*/

33 }

34 tail->next=NULL; /\* 对指针域赋NULL值 \*/

35 }

36 headp=loc\_head; /\* 使头指针headp指向新创建的链表 \*/

37 }

**解答：**

1. 错误原因：creat\_list函数中传入的是一级指针，应该传入二级指针

第7行应改为void create\_list (struct s\_list \*\*headp,int \*p);

第12行应改为create\_list(&head,s);

第20行应改为void create\_list(struct s\_list \*\*headp,int \*p)

第36行应改为\*headp=loc\_head;

1. 错误修改后运行结果：

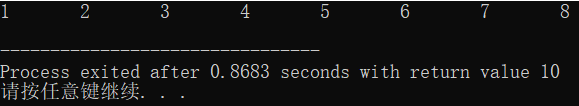


图7-1源程序修改替换题（1）运行结果

（2）修改替换create\_list函数，将其建成一个后进先出的链表，后进先出链表的头指针始终指向最后创建的结点（链头），后建结点指向先建结点，先建结点始终是尾结点。

**解答：**

1. 修改后creat\_list函数：

void create\_list(struct s\_list \*\*headp,int \*p)

{

struct s\_list \* loc\_head=NULL,\*tail,\*SP;

if(p[0]==0)

;

else {

loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));

loc\_head->data=\*p++;

tail=loc\_head;

tail->next=NULL;

while(\*p){

SP=(struct s\_list \*)malloc(sizeof(struct s\_list));

SP->data=\*p++;

SP->next=loc\_head;

loc\_head=SP;

}

}

\*headp=loc\_head;

}

1. 替换后运行结果：

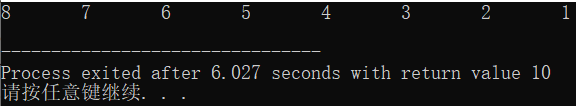


图7-2源程序修改替换题（2）运行结果

**7.2.3编程设计题**

（1）设计一个字段结构struct bits，它将一个8位无符号字节从最低位向最高位声明为8个字段，各字段依次为bit0, bit1, …, bit7，且bit0的优先级最高。同时设计8个函数，第i个函数以biti(i=0,1,2,…,7)为参数，并且在函数体内输出biti的值。将8个函数的名字存入一个函数指针数组p\_fun。如果bit0为1，调用p\_fun[0]指向的函数。如果struct bits中有多位为1，则根据优先级从高到低依次调用函数指针数组p\_fun中相应元素指向的函数。8个函数中的第0个函数可以设计为：

void f0(struct bits b)

{

printf(“the function %d is called!\n”,b);

}

**解答：**

1. 算法流程如图7-3所示：

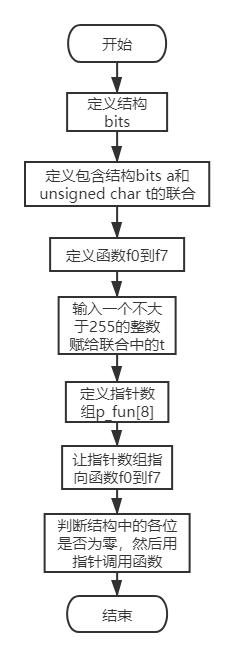


图7-3编程题1流程图

1. 源程序清单：

#include<stdio.h>

struct bits{

unsigned char bit0 : 1;

unsigned char bit1 : 1;

unsigned char bit2 : 1;

unsigned char bit3 : 1;

unsigned char bit4 : 1;

unsigned char bit5 : 1;

unsigned char bit6 : 1;

unsigned char bit7 : 1;

};

union w{

struct bits a;

unsigned char t;

}m;

void f0(struct bits b)

{

printf("the 0 function %d is called!\n",b.bit0);

}

void f1(struct bits b)

{

printf("the 1 function %d is called!\n",b.bit1);

}

void f2(struct bits b)

{

printf("the 2 function %d is called!\n",b.bit2);

}

void f3(struct bits b)

{

printf("the 3 function %d is called!\n",b.bit3);

}

void f4(struct bits b)

{

printf("the 4 function %d is called!\n",b.bit4);

}

void f5(struct bits b)

{

printf("the 5 function %d is called!\n",b.bit5);

}

void f6(struct bits b)

{

printf("the 6 function %d is called!\n",b.bit6);

}

void f7(struct bits b)

{

printf("the 7 function %d is called!\n",b.bit7);

}

int main()

{

printf("请输入一个不大于255的正整数：");

scanf("%ud",&m.t);

void (\*p\_fun[8])(struct bits b);

p\_fun[0] = f0;

p\_fun[1] = f1;

p\_fun[2] = f2;

p\_fun[3] = f3;

p\_fun[4] = f4;

p\_fun[5] = f5;

p\_fun[6] = f6;

p\_fun[7] = f7;

if (m.a.bit0) p\_fun[0](m.a);

if (m.a.bit1) p\_fun[1](m.a);

if (m.a.bit2) p\_fun[2](m.a);

if (m.a.bit3) p\_fun[3](m.a);

if (m.a.bit4) p\_fun[4](m.a);

if (m.a.bit5) p\_fun[5](m.a);

if (m.a.bit6) p\_fun[6](m.a);

if (m.a.bit7) p\_fun[7](m.a);

return 0;

}

1. 测试：
2. 测试数据：
3. 对应测试数据运行结果截图：

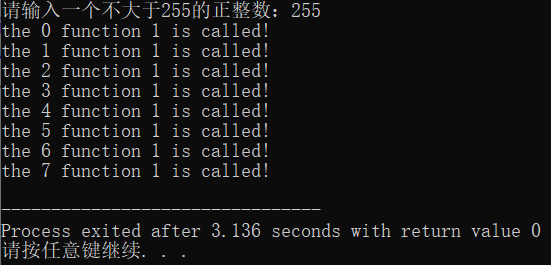


图7-4编程题1测试用例运行结果

（2）用单向链表建立一张班级成绩单，包括每个学生的学号、姓名、英语、高等数学、普通物理、C语言程序设计四门课程的成绩。用函数编程实现下列功能：

(1) 输入每个学生的各项信息。

(2) 输出每个学生的各项信息。

(3) 修改指定学生的指定数据项的内容。

(4) 统计每个同学的平均成绩（保留2位小数）。

(5) 输出各位同学的学号、姓名、四门课程的总成绩和平均成绩。

**解答：**

1. 算法流程如图7-5所示：

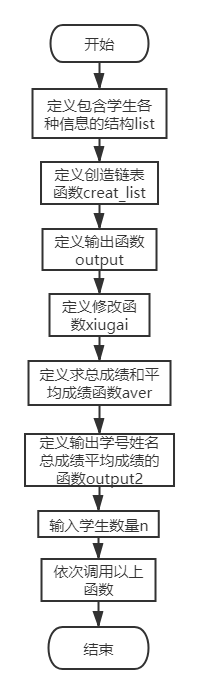


图7-5编程题2流程图

1. 源程序清单：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct list{

char num[8];

char name[9];

int english;

int math;

int phy;

int c;

struct list \*next;

}\*p,\*q;

struct list \*creat\_list(int n)

{

int i;

struct list \*head = NULL;

for (i = 0; i < n; i++)

{

p = (struct list \*)malloc(sizeof(struct list));

if (i == 0) head=p;

else q->next=p;

printf("请输入学号: ");

scanf("%s", p->num);

printf("请输入姓名: ");

scanf("%s", p->name);

printf("请输入英语 数学 物理 C语言成绩：");

scanf("%d%d%d%d", &p->english, &p->math, &p->phy, &p->c);

printf("\n");

q = p;

p->next = NULL;

}

return head;

}

void output(struct list \*head, int n)

{

printf("学号\t姓名\t 英语\t 数学\t 物理\tC 语言\n");

for (p = head; n > 0; n--, p = p->next)

{

printf("%-5s %3s %8d %9d %6d %6d\n", p->num, p->name, p->english, p->math, p->phy, p->c);

}

}

void xiugai(struct list \*head)

{

char a[8];

int s;

int k, i, m;

printf("你想修改第几个学生的信息？\n");

scanf("%d", &k);

for (q = head, i = 0; i < k - 1; i++)

q = q->next;

printf("你想修改哪项信息？\n");

printf("1:学号 2:姓名 3:英语 4:数学 5:物理 6:C语言\n");

scanf("%d", &m);

switch (m)

{

case 1:printf("输入新学号:");

scanf("%s", a);

strcpy(q->num, a);

break;

case 2:printf("输入新姓名:");

scanf("%s", a);

strcpy(q->name, a);

break;

case 3:printf("输入新英语成绩:");

scanf("%d", &s);

q->english = s;

break;

case 4:printf("输入新数学成绩:");

scanf("%d", &s);

q->math = s;

break;

case 5:printf("输入新物理成绩:");

scanf("%d", &s);

q->phy = s;

break;

case 6:printf("输入新c语言成绩:");

scanf("%d", &s);

q->c = s;

break;

default:printf("error!\n");

}

printf("\n");

}

void aver(struct list \*head, int n,double \*all,double \*ave)

{

int i;

q = head;

for (i = 0; i < n; i++)

{

all[i]= (q->english + q->math + q->phy + q->c);

ave[i]=all[i]/4.0;

q = q->next;

}

}

void output2(struct list \*head, int n,double \*all,double \*ave)

{

int i=0;

printf("学号\t姓名\t总成绩\t 平均成绩\n");

for (p = head; n > 0; n--, p = p->next,i++)

{

printf("%-5s %5s %8.1f %9.2f \n", p->num, p->name, all[i], ave[i]);

}

}

void main()

{

int n;

struct list \*head;

printf("输入学生数量: ");

scanf("%d", &n);

head = creat\_list(n);

output(head, n);

xiugai(head);

printf("\n");

double all[n],ave[n];

aver(head, n,all,ave);

output2(head, n,all,ave);

}

1. 测试：
2. 测试数据：

3

001

小明

100 100 99 98

002

小王

99 99 95 98

003

小红

100 99 99 99

2

4

95

1. 对应测试数据运行结果截图：

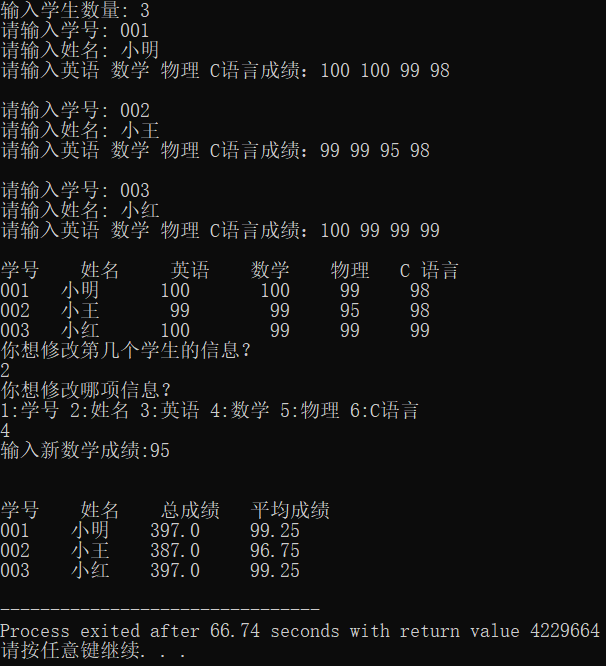


图7-6编程题1测试用例运行结果

**7.2.4选做题**

（1）对编程设计题第（2）题的程序，增加按照平均成绩进行升序排序的函数，写出用交换结点数据域的方法升序排序的函数，排序可用选择法或冒泡法。

**解答：**

1. 算法流程如图7-7所示：

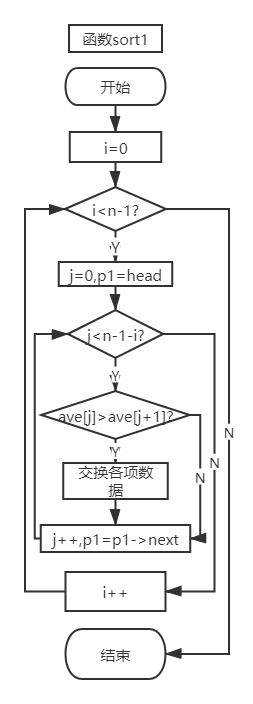


图7-7 选做题1流程图

1. 函数sort1清单：

void sort1(int n,double \*all,double \*ave,struct list \*head)

{

struct list \*p1,\*p2;

int i,j,w;

double t;

char a[9];

for(i=0;i<n-1;i++)

for(j=0,p1=head;j<n-1-i;j++,p1=p1->next)

{

if(ave[j]>ave[j+1])

{

p2=p1->next;

t=ave[j];ave[j]=ave[j+1];ave[j+1]=t;

t=all[j];all[j]=all[j+1];all[j+1]=t;

strcpy(a,p1->name);

strcpy(p1->name,p2->name);

strcpy(p2->name,a);

strcpy(a,p1->num);

strcpy(p1->num,p2->num);

strcpy(p2->num,a);

w=p1->c;p1->c=p2->c;p2->c=w;

w=p1->english;p1->english=p2->english;p2->english=w;

w=p1->math;p1->math=p2->math;p2->math=w;

w=p1->phy;p1->phy=p2->phy;p2->phy=w;

}

}

}

1. 测试：

对应测试数据排序后运行结果截图：



图7-8选做题1运行结果

（2）对选做题第（1）题，进一步写出用交换结点指针域的方法升序排序的函数。

**解答：**

1. 函数sort2清单：

void sort2(int n,double \*all,double \*ave,struct list \*\*headp)

{

struct list \*p1,\*p2,\*p3;

int i,j;

double t;

for(i=0;i<n-1;i++)

for(j=0,p1=\*headp,p3=\*headp;j<n-1-i;j++,p1=p1->next,p3=p3->next)

{

if(j==1)p3=\*headp;

if(ave[j]>ave[j+1])

{

p2=p1->next;

t=ave[j];ave[j]=ave[j+1];ave[j+1]=t;

t=all[j];all[j]=all[j+1];all[j+1]=t;

if(j==0)

{

p3=p2->next;

p2->next=p1;

p1->next=p3;

\*headp=p2;

}

else

{

p3->next=p2;

p1->next=p2->next;

p2->next=p1;

}

}

}

}

1. 测试：

对应测试数据排序后运行结果截图：

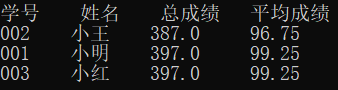


图7-9选做题2运行结果

（3）采用双向链表重做编程设计题第（2）题。

**解答：**

1. 源程序清单：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct list{

char num[8];

char name[9];

int english;

int math;

int phy;

int c;

struct list \*prior;

struct list \*next;

}\*p,\*q;

struct list \*creat\_list(int n)

{

int i;

struct list \*head = NULL;

for (i = 0; i < n; i++)

{

p = (struct list \*)malloc(sizeof(struct list));

if (i == 0) head=p;

else q->next=p,p->prior=q;

printf("请输入学号: ");

scanf("%s", p->num);

printf("请输入姓名: ");

scanf("%s", p->name);

printf("请输入英语 数学 物理 C语言成绩：");

scanf("%d%d%d%d", &p->english, &p->math, &p->phy, &p->c);

printf("\n");

q = p;

p->next = NULL;

}

p->next=head;

head->prior=p;

return head;

}

void output(struct list \*head, int n)

{

printf("学号\t姓名\t 英语\t 数学\t 物理\tC 语言\n");

for (p = head; n > 0; n--, p = p->next)

{

printf("%-5s %3s %8d %9d %6d %6d\n", p->num, p->name, p->english, p->math, p->phy, p->c);

}

}

void xiugai(struct list \*head)

{

char a[8];

int s;

int k, i, m;

printf("你想修改第几个学生的信息？\n");

scanf("%d", &k);

for (q = head, i = 0; i < k - 1; i++)

q = q->next;

printf("你想修改哪项信息？\n");

printf("1:学号 2:姓名 3:英语 4:数学 5:物理 6:C语言\n");

scanf("%d", &m);

switch (m)

{

case 1:printf("输入新学号:");

scanf("%s", a);

strcpy(q->num, a);

break;

case 2:printf("输入新姓名:");

scanf("%s", a);

strcpy(q->name, a);

break;

case 3:printf("输入新英语成绩:");

scanf("%d", &s);

q->english = s;

break;

case 4:printf("输入新数学成绩:");

scanf("%d", &s);

q->math = s;

break;

case 5:printf("输入新物理成绩:");

scanf("%d", &s);

q->phy = s;

break;

case 6:printf("输入新c语言成绩:");

scanf("%d", &s);

q->c = s;

break;

default:printf("error!\n");

}

printf("\n");

}

void aver(struct list \*head, int n,double \*all,double \*ave)

{

int i;

q = head;

for (i = 0; i < n; i++)

{

all[i]= (q->english + q->math + q->phy + q->c);

ave[i]=all[i]/4.0;

q = q->next;

}

}

void output2(struct list \*head, int n,double \*all,double \*ave)

{

int i=0;

printf("学号\t姓名\t总成绩\t 平均成绩\n");

for (p = head; n > 0; n--, p = p->next,i++)

{

printf("%-5s %5s %8.1f %9.2f \n", p->num, p->name, all[i], ave[i]);

}

}

int main()

{

int n;

struct list \*head;

printf("输入学生数量: ");

scanf("%d", &n);

head = creat\_list(n);

output(head, n);

xiugai(head);

printf("\n");

double all[n],ave[n];

aver(head, n,all,ave);

output2(head, n,all,ave);

return 0;

}

1. 测试：
2. 测试数据：

3

001

小明

100 100 99 98

002

小王

99 99 95 98

003

小红

100 99 99 99

2

4

95

1. 对应测试数据运行结果截图：

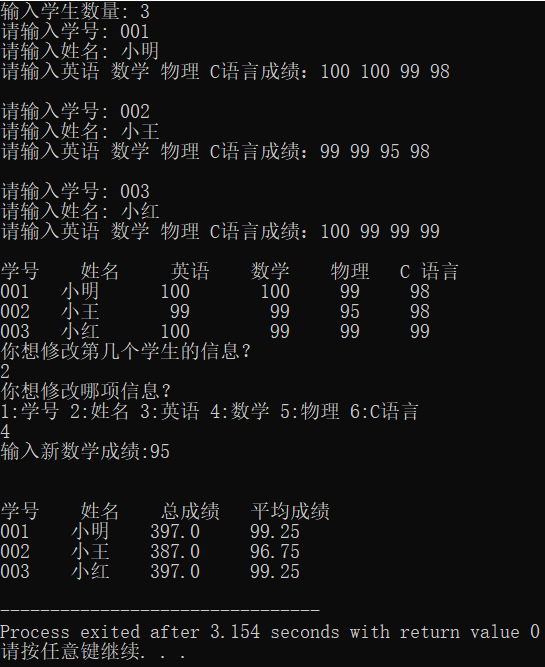


图7-10选做题3测试用例运行结果

**7.3实验小结**

用逗号连接的多个表达式的值为后一个逗号后表达式的值； 对指针进行加减操作时一定注意该指针类型。

**参考文献**

[1] 曹计昌,卢萍,李开. C语言程序设计,北京： 科学出版社,2013

[2] 李开,卢萍,曹计昌. C语言实验与课程设计, 北京：科学出版社,2011