

**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：**

**软件学院**

**目 录**

[7 结构与联合实验 1](#_Toc185535892)

[7.1 实验目的 1](#_Toc185535893)

[7.2 实验题目及要求 1](#_Toc185535894)

[7.3 实验小结 18](#_Toc185535895)

实验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 实验题目及要求

**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 |

**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：

#include "stdio.h"

#include "stdlib.h"

struct s\_list{

int data; /\* 数据域 \*/

struct s\_list \*next; /\* 指针域 \*/

} ;

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

void main(void)

{

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

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

create\_list(head,s); /\* 创建新链表 \*/

p=head; /\*遍历指针p指向链头 \*/

while(p){

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

p=p->next; /\*遍历指针p指向下一结点 \*/

}

printf("\n");

}

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

{

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

if(p[0]==0) /\* 相当于\*p==0 \*/

;

else { /\* loc\_head指向动态分配的第一个结点 \*/

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

loc\_head->data=\*p++; /\* 对数据域赋值 \*/

tail=loc\_head; /\* tail指向第一个结点 \*/

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

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

tail=tail->next; /\* tail指向新创建的结点 \*/

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

}

tail->next=NULL; /\* 对指针域赋NULL值 \*/

}

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

}

**解答：**

* 1. 修改后完整代码

#include "stdio.h"

#include "stdlib.h"

struct s\_list{

int data; /\* 数据域 \*/

struct s\_list \*next; /\* 指针域 \*/

} ;

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

void main(void)

{

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

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

create\_list(&head,s); /\* 创建新链表 \*/

p=head; /\*遍历指针p指向链头 \*/

while(p){

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

p=p->next; /\*遍历指针p指向下一结点 \*/

}

printf("\n");

}

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

{

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

if(p[0]==0) /\* 相当于\*p==0 \*/

;

else { /\* loc\_head指向动态分配的第一个结点 \*/

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

loc\_head->data=\*p++; /\* 对数据域赋值 \*/

tail=loc\_head; /\* tail指向第一个结点 \*/

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

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

tail=tail->next; /\* tail指向新创建的结点 \*/

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

}

tail->next=NULL; /\* 对指针域赋NULL值 \*/

}

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

}

* 1. 修改后运行效果

截图里有图片

描述已自动生成图7-1 程序修改题1的运行结果图

（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. 修改后代码

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

{

struct s\_list \* loc\_head=NULL,\*head;

if(p[0]==0) /\* 相当于\*p==0 \*/

;

else { /\* head指向动态分配的第一个结点 \*/

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

head->data=\*p++; /\* 对数据域赋值 \*/

head->next=NULL;

while(\*p){

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

loc\_head->data=\*p++; /\* 对数据域赋值 \*/

loc\_head->next=head; /\* 新结点指向原链表的头 \*/

head=loc\_head; /\* 新结点成为新的头 \*/

}

}

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

}

**3 程序设计**

以下（1）至（3）题对应Educoder 教学平台“C语言实验”课程，实验7，第17关实验7-1、第18关实验7-2，以及第19关实验7-3。

（1）本关任务：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。用菜单实现下列功能：

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. 源程序清单

#include<stdio.h>

#include <stdlib.h>

struct node {

struct {

int sid;

char name[20];

int english;

int maths;

int physics;

int programming;

} data;

struct node \*next;

};

void inputData(struct node \*\*head) {

int count;

scanf("%d", &count);

struct node \*p = \*head;

for (int i = 0; i < count; i++) {

struct node \*temp = malloc(sizeof(struct node));

scanf("%d %s %d %d %d %d", &temp->data.sid, temp->data.name, &temp->data.english, &temp->data.maths,

&temp->data.physics, &temp->data.programming);

temp->next = NULL;

if (p) {

p->next = temp;

p = p->next;

} else {

p = temp;

\*head = p;

}

}

printf("完成了%d位同学的成绩输入。\n", count);

}

void exportData(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %d %d %d %d\n", p->data.sid, p->data.name, p->data.english, p->data.maths,

p->data.physics, p->data.programming);

p = p->next;

}

}

void modifyData(struct node \*head) {

int sid, target;

scanf("%d %d", &sid, &target);

struct node \*p = head;

while (p) {

if (p->data.sid == sid) {

switch (target) {

case 1:

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

break;

case 2:

scanf("%d", &p->data.english);

break;

case 3:

scanf("%d", &p->data.maths);

break;

case 4:

scanf("%d", &p->data.physics);

break;

case 5:

scanf("%d", &p->data.programming);

break;

}

printf("%d %s %d %d %d %d\n", p->data.sid, p->data.name, p->data.english, p->data.maths,

p->data.physics, p->data.programming);

return;

}

p = p->next;

}

}

void showAverage(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %.2f\n", p->data.sid, p->data.name, (p->data.english + p->data.maths +

p->data.physics + p->data.programming) / 4.0);

p = p->next;

}

}

void showTotalAndAverage(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %d %.2f\n", p->data.sid, p->data.name, p->data.english + p->data.maths +

p->data.physics + p->data.programming,

(p->data.english + p->data.maths + p->data.physics + p->data.programming) / 4.0);

p = p->next;

}

}

int main() {

struct node \*head = NULL;

int opcode;

scanf("%d", &opcode);

while (opcode) {

switch (opcode) {

case 1:

inputData(&head);

break;

case 2:

exportData(head);

break;

case 3:

modifyData(head);

break;

case 4:

showAverage(head);

break;

case 5:

showTotalAndAverage(head);

break;

}

scanf("%d", &opcode);

}

return 0;

}

1. 测试数据

表7-1 编程题1的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 程序输入 | 理 论 结 果 | 运 行 结 果 |
| 用例1 | 1  5  2021001 Jack 90 92 87 95  2021002 Mike 85 70 75 90  2021003 Joe 77 86 90 75  2021004 Andy 95 97 92 95  2021005 Rose 90 87 88 89  3  2021003 2 96 | 完成了5位同学的成绩输入。  2021003 Joan 77 96 90 75 | 完成了5位同学的成绩输入。  2021003 Joan 77 96 90 75 |

1. 文本

   描述已自动生成运行效果

图7-2 编程题1的运行结果图

（2）本关任务：对程序设计题第（1）题的程序，⑥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. 源程序清单

#include<stdio.h>

#include <stdlib.h>

struct data {

int sid;

char name[20];

int english;

int maths;

int physics;

int programming;

};

struct node {

struct data data;

struct node \*next;

};

int count;

void inputData(struct node \*\*head) {

scanf("%d", &count);

struct node \*p = \*head;

for (int i = 0; i < count; i++) {

struct node \*temp = malloc(sizeof(struct node));

scanf("%d %s %d %d %d %d", &temp->data.sid, temp->data.name, &temp->data.english, &temp->data.maths,

&temp->data.physics, &temp->data.programming);

temp->next = NULL;

if (p) {

p->next = temp;

p = p->next;

} else {

p = temp;

\*head = p;

}

}

printf("完成了%d位同学的成绩输入。\n", count);

}

void exportData(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %d %d %d %d\n", p->data.sid, p->data.name, p->data.english, p->data.maths,

p->data.physics, p->data.programming);

p = p->next;

}

}

void modifyData(struct node \*head) {

int sid, target;

scanf("%d %d", &sid, &target);

struct node \*p = head;

while (p) {

if (p->data.sid == sid) {

switch (target) {

case 1:

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

break;

case 2:

scanf("%d", &p->data.english);

break;

case 3:

scanf("%d", &p->data.maths);

break;

case 4:

scanf("%d", &p->data.physics);

break;

case 5:

scanf("%d", &p->data.programming);

break;

}

printf("%d %s %d %d %d %d\n", p->data.sid, p->data.name, p->data.english, p->data.maths,

p->data.physics, p->data.programming);

return;

}

p = p->next;

}

}

void showAverage(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %.2f\n", p->data.sid, p->data.name, (p->data.english + p->data.maths +

p->data.physics + p->data.programming) / 4.0);

p = p->next;

}

}

void showTotalAndAverage(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %d %.2f\n", p->data.sid, p->data.name, p->data.english + p->data.maths +

p->data.physics + p->data.programming,

(p->data.english + p->data.maths + p->data.physics + p->data.programming) / 4.0);

p = p->next;

}

}

int getTotal(struct node \*p) {

return p->data.english + p->data.maths + p->data.physics + p->data.programming;

}

struct node \*findMinimum(struct node \*head) {

struct node \*p = head->next;

struct node \*min = head;

while (p) {

if (getTotal(p) < getTotal(min)) {

min = p;

}

p = p->next;

}

return min;

}

void selectionSort(struct node \*head) {

struct node \*p = head;

for (int i = 0; i < count; i++) {

struct node \*min = findMinimum(p);

if (p != min) {

struct data temp = p->data;

p->data = min->data;

min->data = temp;

}

p = p->next;

}

}

int main() {

struct node \*head = NULL;

int opcode;

scanf("%d", &opcode);

while (opcode) {

switch (opcode) {

case 1:

inputData(&head);

break;

case 2:

exportData(head);

break;

case 3:

modifyData(head);

break;

case 4:

showAverage(head);

break;

case 5:

showTotalAndAverage(head);

break;

case 6:

selectionSort(head);

showAverage(head);

break;

}

scanf("%d", &opcode);

}

return 0;

}

1. 测试数据

表7-2 编程题2的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 程序输入 | 理 论 结 果 | 运 行 结 果 |
| 用例1 | 1  5  2021001 Jack 90 92 87 95  2021002 Mike 85 70 75 90  2021003 Joe 77 86 90 75  2021004 Andy 95 97 92 95  2021005 Rose 90 87 88 89  6 | 完成了5位同学的成绩输入。  2021002 Mike 80.00  2021003 Joe 82.00  2021005 Rose 88.50  2021001 Jack 91.00  2021004 Andy 94.75 | 完成了5位同学的成绩输入。  2021002 Mike 80.00  2021003 Joe 82.00  2021005 Rose 88.50  2021001 Jack 91.00  2021004 Andy 94.75 |

1. 文本

   低可信度描述已自动生成运行效果

图7-3 编程题2的运行结果图

（3）本关任务：对程序设计题第（2）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. 源程序清单

#include<stdio.h>

#include <stdlib.h>

struct data {

int sid;

char name[20];

int english;

int maths;

int physics;

int programming;

};

struct node {

struct data data;

struct node \*next;

};

int count;

void inputData(struct node \*\*head) {

scanf("%d", &count);

struct node \*p = \*head;

for (int i = 0; i < count; i++) {

struct node \*temp = malloc(sizeof(struct node));

scanf("%d %s %d %d %d %d", &temp->data.sid, temp->data.name, &temp->data.english, &temp->data.maths,

&temp->data.physics, &temp->data.programming);

temp->next = NULL;

if (p) {

p->next = temp;

p = p->next;

} else {

p = temp;

\*head = p;

}

}

printf("完成了%d位同学的成绩输入。\n", count);

}

void exportData(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %d %d %d %d\n", p->data.sid, p->data.name, p->data.english, p->data.maths,

p->data.physics, p->data.programming);

p = p->next;

}

}

void modifyData(struct node \*head) {

int sid, target;

scanf("%d %d", &sid, &target);

struct node \*p = head;

while (p) {

if (p->data.sid == sid) {

switch (target) {

case 1:

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

break;

case 2:

scanf("%d", &p->data.english);

break;

case 3:

scanf("%d", &p->data.maths);

break;

case 4:

scanf("%d", &p->data.physics);

break;

case 5:

scanf("%d", &p->data.programming);

break;

}

printf("%d %s %d %d %d %d\n", p->data.sid, p->data.name, p->data.english, p->data.maths,

p->data.physics, p->data.programming);

return;

}

p = p->next;

}

}

void showAverage(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %.2f\n", p->data.sid, p->data.name, (p->data.english + p->data.maths +

p->data.physics + p->data.programming) / 4.0);

p = p->next;

}

}

void showTotalAndAverage(struct node \*head) {

struct node \*p = head;

while (p) {

printf("%d %s %d %.2f\n", p->data.sid, p->data.name, p->data.english + p->data.maths +

p->data.physics + p->data.programming,

(p->data.english + p->data.maths + p->data.physics + p->data.programming) / 4.0);

p = p->next;

}

}

int getTotal(struct node \*p) {

return p->data.english + p->data.maths + p->data.physics + p->data.programming;

}

struct node \*findMinimum(struct node \*head) {

struct node \*p = head->next;

struct node \*min = head;

while (p) {

if (getTotal(p) < getTotal(min)) {

min = p;

}

p = p->next;

}

return min;

}

void selectionSort(struct node \*head) {

struct node \*p = head;

for (int i = 0; i < count; i++) {

struct node \*min = findMinimum(p);

if (p != min) {

struct data temp = p->data;

p->data = min->data;

min->data = temp;

}

p = p->next;

}

}

void selectionSort2(struct node \*\*head) {

struct node \*p = \*head;

for (int i = 0; i < count; i++) {

struct node \*min = findMinimum(p);

if (p != min) {

struct node \*prev = NULL, \*curr = \*head;

while (curr != min) {

prev = curr;

curr = curr->next;

}

if (prev) {

prev->next = min->next;

}

min->next = p;

if (p == \*head) {

\*head = min;

} else {

struct node \*temp = \*head;

while (temp->next != p) {

temp = temp->next;

}

temp->next = min;

}

}

p = \*head;

for (int j = 0; j < i; j++)

p = p->next;

}

}

int main() {

struct node \*head = NULL;

int opcode;

scanf("%d", &opcode);

while (opcode) {

switch (opcode) {

case 1:

inputData(&head);

break;

case 2:

exportData(head);

break;

case 3:

modifyData(head);

break;

case 4:

showAverage(head);

break;

case 5:

showTotalAndAverage(head);

break;

case 6:

selectionSort2(&head);

showAverage(head);

break;

}

scanf("%d", &opcode);

}

return 0;

}

1. 测试数据

表7-3 编程题3的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试  用例 | 程序输入 | 理 论 结 果 | 运 行 结 果 |
| 用例1 | 1  5  2021001 Jack 90 92 87 95  2021002 Mike 85 70 75 90  2021003 Joe 77 86 90 75  2021004 Andy 95 97 92 95  2021005 Rose 90 87 88 89  6 | 完成了5位同学的成绩输入。  2021002 Mike 80.00  2021003 Joe 82.00  2021005 Rose 88.50  2021001 Jack 91.00  2021004 Andy 94.75 | 完成了5位同学的成绩输入。  2021002 Mike 80.00  2021003 Joe 82.00  2021005 Rose 88.50  2021001 Jack 91.00  2021004 Andy 94.75 |

1. 文本

   描述已自动生成运行效果

图7-4 编程题3的运行结果图

7.3 实验小结

1. 在本次实验中，我学会了链表的数据结构，并且体验了链表的创建、遍历、删除节点等操作。
2. 在本次实验中，调试功能有助于我直观的查看链表内数据的变化，使程序查错更加便捷高效。