武汉纺织大学

数学与计算机学院

数据结构实验报告

2019 ~2020学年第一学期

|  |  |
| --- | --- |
| 专 业 班 级 | 物联网11802班 |
| 学生学号 | 1802220113 |
| 学生姓名 | 梁少伟 |
| 任 课 教 师 | 叶鹏 |
| 实验指导教师 | 叶鹏 |
| 实验地点 | 实训楼 |
| 填写时间 | 2019/10/18 |
|  | |

**填写说明**

1. 报告需要学生填写的内容包括：实验要求、完整的实验结果记录、实验结果分析、实验总结。实验评价由教师填写。
2. 各部分填写内容的要求
   * **实验要求**主要填写源代码。
   * **实验结果记录**中主要反映程序运行后的效果。
   * **实验结果分析**中主要就程序运行结果是否符合预期进行分析，对于达不到预期的分析其产生的原因。
   * **实验总结**中主要阐述在实验中遇到了哪些问题，如何解决的。
3. 实验报告及源代码的提交方式
   * 报告和源代码一并发送到trustie平台。
   * 附件中应有两个文件：一是word版实验报告，**实验报告的命名规则**是：“学号\_姓名\_班级\_实验X”。另一个是压缩文件。它是将整个工程压缩成一个文件，这个**压缩文件的命名规则**是：“学号\_姓名\_班级\_实验X\_源代码”。

**实验3 一元多项式加法的实现**

**一、实验目的**

1.掌握使用VC++上机调试线性表的基本方法；

2.掌握一元多项式加法的实现。

二、**实验内容**

完成2个一元多项式的加法，必须输入不少于两组数据进行测试。

**三、实验要求**

1.认真阅读和理解本实验的算法。

2.上机运行写出的程序，并且独立调试通过。

（源程序）

**1．“head.h”头文件**

#include"stdio.h"

#include"malloc.h"

#include"iostream"

using namespace std;

typedef int datatype1;

typedef int datatype2;

typedef struct node

{

datatype2 coef;

datatype1 expn;

struct node\* next;

}Slist;

**2.“function.cpp”功能函数**

#include"head.h"

Slist\* CreateList()

{

Slist\* head, \* s, \* r;

float m;

int n;

head = (Slist\*)malloc(sizeof(Slist));

head = NULL;

s = NULL;

cout << "请输入系数m和指数n: ";

cin >> m >> n;

while (m != 0 )

{

r=(Slist\*)malloc(sizeof(Slist));

r->coef = m;

r->expn = n;

if (head == NULL)

head = r;

else

s->next = r;

s = r;

cout << "请输入系数m和指数n：" ;

cin >> m >> n;

}

if (s->next != NULL)

s->next = NULL; /////////////////////////////函数的初始化（未进行次序排序）

node\* p, \* q; /////////////////////////////函数进行内部排序

int num = 0, j = 0;

q = head;

while (q != NULL)

{

q = q->next;

num++;

}//获取链表的长度

for (int i = 0; i < num - 1; i++)////////冒泡排序的基本思路

{

p = q = head;

j = num - i - 1; //减少每一趟循环中两两比较的次数

while (p->next != NULL && j != 0)

{

j--;

if (p->expn > p->next->expn)

{

//节点的交换

if (p == head) head = p->next;

else q->next = p->next;

q->next = p->next;

q = q->next;

p->next = q->next;

q->next = p;

//执行完上面的过程后，为了能够让p顺利地执行移动到交换后的下一位 .

p = q;

}

q = p; //为了能让q保持在p的前面

p = p->next; //p指针后移，即p变成了在q的前面

}

}

return head;

}

void ShowSList(Slist\* head)

{

Slist\* p;

p = head;

while (p != NULL)

{

if (p->next == NULL)

{

cout << p->coef << "X^" << p->expn<<endl;

break;

}

else

{

cout << p->coef << "X^" << p->expn << " + ";

p = p->next;

}

}

}

Slist \* Listadd(node\* head1, node\* head2)

{

node\* p, \* q, \* pre, \* temp,\*head3;

p = head1;

q = head2;

pre = (Slist\*)malloc(sizeof(Slist));

//// head3=(Slist\*)malloc(sizeof(Slist));

head3=pre;

while(p!=NULL && q!=NULL)

{

if(p->expn<q->expn)

{

temp= (Slist\*)malloc(sizeof(Slist));

temp->expn=p->expn;

temp->coef=p->coef;

pre->next=temp;

p=p->next;

pre=pre->next;

pre->next = NULL;

}

else if(p->expn >q->expn)

{

temp= (Slist\*)malloc(sizeof(Slist));

temp->expn=q->expn;

temp->coef=q->coef;

pre->next=temp;

q=q->next;

pre=pre->next;

pre->next = NULL;

}

else

{

temp= (Slist\*)malloc(sizeof(Slist));

temp->coef = p->coef+q->coef;

temp->expn=p->expn;

pre->next=temp;

q=q->next;

p=p->next;

pre=pre->next;

pre->next = NULL;

}

}

if(p==NULL)

{pre->next=q;}

else

{pre->next=p;}

return head3->next;

}

**3.“main.cpp”主函数**

#include"head.h"

void ShowSList(Slist\* head);///打印

Slist\* CreateList();///函数的创建

Slist \* Listadd(node\* head1, node\* head2);

int main()

{

Slist\* head1,\*head2,\*head3;

head1= CreateList();

cout << "所输入函数F1为：" << endl;

ShowSList(head1);

cout << "===================================="<<endl;

head2 = CreateList();

cout << "所输入函数F2为：" << endl;

cout << "===================================="<<endl;

ShowSList(head2);

cout << "合并函数F3为：" << endl;

head3 = Listadd(head1, head2);

ShowSList(head3);

return 0;

}

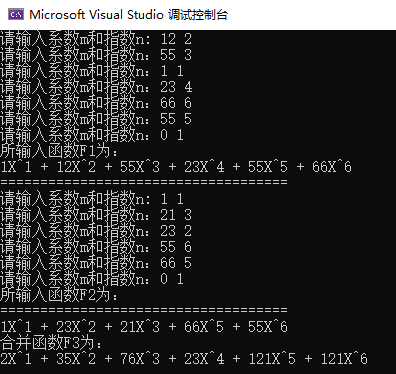
**四．完整的实验结果记录**

**第一次试验 （赋值时无规则）：**

**赋值为：F1=1X^1+12X^2+55X^3+23X^4+55X^5+66X^6**

**F2=1X^1+23X^2+21X^3+66X^5+55X^6**

**合并函数为：F3=2X^1+35X^2+76X^3+23X^4+121X^5+121X^6**

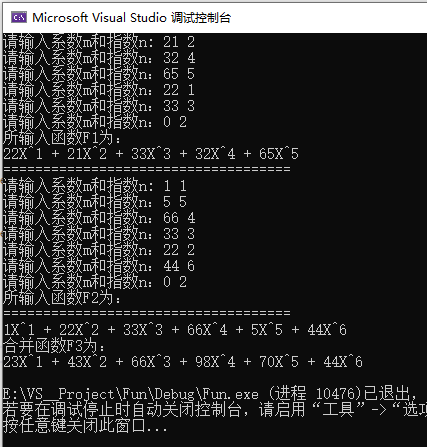


**第二次试验（赋值时无规则）**

**赋值为：F1=22X^1+21X^2+33X^3+32X^4+65X^5**

**F2=1X^1+22X^2+33X^3+66X^4+5X^5+44X^6**

**合并函数为：F3=23X^1+43X^2+66X^3+98X^4+70X^5+44X^6**

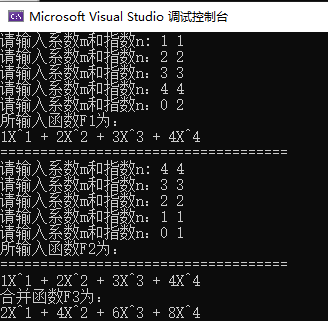


**第三次试验（赋值时无规则）**

**赋值为：F1=1X^1+2X^2+3X^3+4X^4**

**F2=1X^1+2X^2+3X^3+4X^4**

**合并函数为：F3=2X^1+4X^2+6^3+8X^4**



1. **实验结果分析**

本次实验均符合条件，都能正确的输出。

1. **实验总结**

**通过这次实验，感觉自己的编程能力有所提高，调试的时候错误出现的少了，这次实验的难处主要是合并的时候第一个总是乱码，一直没调对，最后才发现自己建立F1 F2时没有空头结点，而我的合并函数有空头结点，导致打印的时候第一个一直是乱码。最后函数返回值变为合并函数F3的next才搞对。**

**七、实验评价**（教师）