**数据结构**

**——实验报告书**



姓名：樊雨晨

学号：201632110111

班级：软件工程(中外)16 班

学院：数理与信息工程学院

日期：2017年05月10日

# **实验二 链表及其多项式相加**

## **问题**

通过有序对输入多项式的各个项，利用单链表存储该一元多项式，并建立的2个存储一元多项式的单链表，然后完成2个一元多项式的相加，并输出相加后的多项式。

## **实验思路**

1. 两个链，将第二个链的数据加入第一个链。
2. 三个链，前两个链的数据通过计算后加到第三个链。

## **实验步骤**

1. 首先创建两个或三个链表。
2. 对两个链表进行赋值。
3. 然后将链表的数据进行比较，是否进行计算通过链表中的指数决定，计算后合并入第一个链表或者第三个链表。
4. 打印计算后的链表。

## **代码**

（1）#**include** <iostream>

#**include** <list>

#**include** <numeric>

#**include** <algorithm>

**using** **namespace** std;

**typedef** **struct**{

**int** coefficient;

**int** index;

}Polynomial;

**typedef** list<Polynomial> INTLIST;

**int** **compare**(**int** a,**int** b)

{

**int** answer;

**if**(a<b)

{

answer=-1;

}

**else** **if**(a>b)

{

answer=1;

}

**else**

{

answer=0;

}

**return** answer;

}

**int** **main**()

{

INTLIST L1;

INTLIST L2;

INTLIST L3;

//////addelements///////

Polynomial Temp;

**int** a,b;

**int** i;

scanf("%d",&a);

**for**(i=0;i<a;i++)

{

scanf("%d%d",&Temp.coefficient,&Temp.index);

L1.push\_back(Temp);

}

scanf("%d",&b);

**for**(i=0;i<b;i++)

{

scanf("%d%d",&Temp.coefficient,&Temp.index);

L2.push\_back(Temp);

}

//////calculation/////

INTLIST::iterator iter1;

INTLIST::iterator iter2;

INTLIST::iterator iter3;

**int** sum;

iter1=L1.begin();

iter2=L2.begin();

**while**(iter1!=L1.end()&&iter2!=L2.end())

{

**int** x,y;

x=iter1->index;

y=iter2->index;

**switch**(compare(x,y))

{

**case** -1: L3.push\_back(\*iter1);iter1++;**break**;

**case** 1: L3.push\_back(\*iter2);iter2++;**break**;

**case** 0: sum=iter1->coefficient+iter2->coefficient;

**if**(sum!=0)

{

iter1->coefficient=sum;

L3.push\_back(\*iter1);

}

iter1++;

iter2++;

**break**;

}

}

**while**(iter1!=L1.end()||iter2!=L2.end())

{

**if**(iter1!=L1.end())

{

L3.push\_back(\*iter1++);

}

**else**

{

L3.push\_back(\*iter2++);

}

}

///////printf///

printf("\n");

INTLIST::iterator iter4;

**for**(iter4 = L3.begin(); iter4 != L3.end(); iter4++)

cout << iter4->coefficient << " " << iter4->index << endl;

}

（2）#**include** <iostream>

#**include** <list>

#**include** <numeric>

#**include** <algorithm>

**using** **namespace** std;

**typedef** **struct**{

**int** coefficient;

**int** index;

}Polynomial;

**typedef** list<Polynomial> INTLIST;

**int** **compare**(**int** a,**int** b)

{

**int** answer;

**if**(a<b)

{

answer=-1;

}

**else** **if**(a>b)

{

answer=1;

}

**else**

{

answer=0;

}

**return** answer;

}

**int** **main**()

{

INTLIST L1;

INTLIST L2;

INTLIST L3;

//////addelements///////

Polynomial Temp;

**int** a,b;

**int** i;

scanf("%d",&a);

**for**(i=0;i<a;i++)

{

scanf("%d%d",&Temp.coefficient,&Temp.index);

L1.push\_back(Temp);

}

scanf("%d",&b);

**for**(i=0;i<b;i++)

{

scanf("%d%d",&Temp.coefficient,&Temp.index);

L2.push\_back(Temp);

}

//////calculation/////

INTLIST::iterator iter1;

INTLIST::iterator iter2;

INTLIST::iterator iter3;

**int** sum;

iter1=L1.begin();

iter2=L2.begin();

**while**(iter1!=L1.end()&&iter2!=L2.end())

{

**int** x,y;

x=iter1->index;

y=iter2->index;

**switch**(compare(x,y))

{

**case** -1: L3.push\_back(\*iter1);iter1++;**break**;

**case** 1: L3.push\_back(\*iter2);iter2++;**break**;

**case** 0: sum=iter1->coefficient+iter2->coefficient;

**if**(sum!=0)

{

iter1->coefficient=sum;

L3.push\_back(\*iter1);

}

iter1++;

iter2++;

**break**;

}

}

**while**(iter1!=L1.end()||iter2!=L2.end())

{

**if**(iter1!=L1.end())

{

L3.push\_back(\*iter1++);

}

**else**

{

L3.push\_back(\*iter2++);

}

}

///////printf///

INTLIST::iterator iter4;

**for**(iter4 = L3.begin(); iter4 != L3.end(); iter4++)

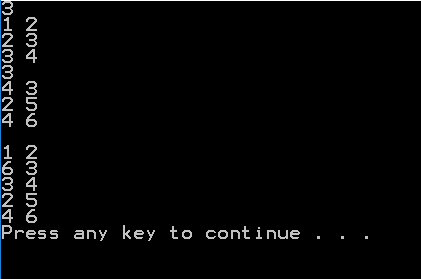
cout << iter4->coefficient << " " << iter4->index << endl;

}

## **测试**

一开始写的都是在主函数外面的函数，没有弄清楚主函数中带入到函数中是值传递，所以运行错误。后来把代码放入主函数变正常。

## **结果**



## **总结**

调用C++list的一些函数，在定义迭代器的时候，迭代器相当于代替链表中的元素，在设置临时的变量，用于输入值代入变量，要注意临时变量要定义为链表元素，而不是链表。还需要加强C++语言学习，比C好用多了=.=>！