# 多项式相乘并对其进行排序

编译环境：xcode

语言：C++

//

// main.cpp

// 多项式相乘

//

// Created by 周子聪 on 16/3/26.

// Copyright © 2016年 独白SKY. All rights reserved.

//

#include <iostream>

using namespace std;

#define Max 20

typedef struct

{

float coef;

int exp;

}polyarray[Max];

struct polynode

{

float coef;

int exp;

polynode \*next;

};

class poly

{

private:

polynode \*Head;

public:

poly()

{

polynode \*null;

null=new polynode;

Head=null;

null->next=NULL;

}

// ~poly();

void createpoly(polyarray a,int n)

{

polynode \*s,\*r;

int i;

r=new polynode;

Head=r;

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

{

s=new polynode;

s->coef=a[i].coef;

s->exp=a[i].exp;

s->next=NULL;

r->next=s;

r=s;

}

}

void polysort()

{

polynode \*p;

p=Head->next;

if(p==NULL||p->next==NULL)

{

cout<<"错误！链表只有一个或没有结点."<<endl;

}

for(polynode \*p1=p;p1->next!=NULL;p1=p1->next)

{

polynode \*p\_min=p1;

for(polynode \*p2=p1->next;p2->next!=NULL;p2=p2->next)

{ if(p2->exp<p\_min->exp)

{

p\_min=p2;

}

}

if(p\_min!=p1)

{

double temp1=p1->coef;

p1->coef=p\_min->coef;

p\_min->coef=temp1;

int temp2=p1->exp;

p1->exp=p\_min->exp;

p\_min->exp=temp2;

}

}

}

void print()

{

polynode\* p=Head->next;

while (p->next!=NULL)

{

if (p->coef!=0) {

cout<<p->coef<<"X^"<<p->exp;

}

if (p->next->next!=NULL)

{

cout<<"+";

}

p=p->next;

}

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

void polymul(poly lb,poly\* result)

{

int iTempExp = 0;

float iTempCoe = 0;

polynode \*Py1 = Head;

polynode \*Py2 = lb.Head;

polynode \*Py3 = result->Head;

while(Py2->next != NULL)

{

Py2 = Py2->next;

Py1 = Head;//每次循环回置头结点

while(Py1->next != NULL)

{

Py1 = Py1->next;

iTempExp = Py1->exp+Py2->exp;

iTempCoe = Py1->coef\*Py2->coef;

if(result->Head->next==NULL)

{

polynode \*temp=new polynode;

temp->exp=iTempExp;

temp->coef=iTempCoe;

temp->next=NULL;

Py3->next=temp;

Py3=Py3->next;

}else

{

polynode\* PNRS = result->Head->next;

while(PNRS != NULL)

{

if(PNRS->exp == iTempExp)

{

PNRS->coef += iTempCoe;

break;

}else if(PNRS->next==NULL)

{

polynode \*temp=new polynode;

temp->exp=iTempExp;

temp->coef=iTempCoe;

temp->next=NULL;

Py3->next=temp;

Py3=Py3->next;

break;

}

PNRS = PNRS->next;

}

}

}

}

}

};

int main()

{

poly la,lb,\*result;

polyarray a={{1.1,6},{2.2,4},{3.3,5},{4.4,3}};

polyarray b={{1.1,4},{3.3,2},{2.2,3}};

la.createpoly (a,4);

lb.createpoly (b,3);

la.polysort ();

lb.polysort ();

cout<<"多项式A为：";

la.print();

cout<<"多项式B为：";

lb.print();

cout<<"多项式A\*B的结果为";

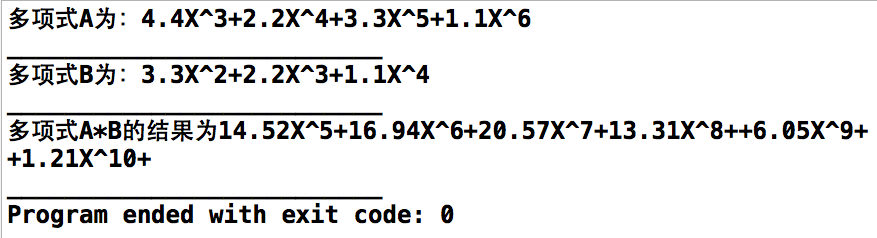
result=new poly;

la.polymul(lb,result);

result->print();

}

# 输出结果（苹果系统控制台）：



参考资料：课本