

一、设计题目

一元多项式的加法、减法、乘法的实现。

二、主要内容

设有一元多项式 $A_m(x)$ 和 $B_n(x)$ 。

$$A_m(x) = A_0 + A_1x + A_2x^2 + A_3x^3 + \cdots + A_mx^m$$

$$B_n(x) = B_0 + B_1x + B_2x^2 + B_3x^3 + \cdots + B_nx^n$$

请实现求 $M(x) = A_m(x) + B_n(x)$ 、 $M(x) = A_m(x) - B_n(x)$ 和 $M(x) = A_m(x) \times B_n(x)$ 。 要求：

- 1) 首先判定多项式是否稀疏
- 2) 采用动态存储结构实现；
- 3) 结果 $M(x)$ 中无重复阶项和无零系数项；
- 4) 要求输出结果的升幂和降幂两种排列情况

三、具体要求及应提交的材料

1. 每个同学以自己的学号和姓名建一个文件夹，如：“”。里面应包括：学生按照课程设计的具体要求所开发的所有源程序（应该放到一个文件夹中）、任务书和课程设计说明书的电子文档。
2. 打印的课程设计说明书（注意：在封面后夹入打印的“任务书”以后再装订）。

四、主要技术路线提示

为把多个小功能结合成一个完整的小软件，需使用“菜单设计”技术（可以是控制台方式下的命令行形式，假设能做成图形方式那么更好）。

五、进度安排

共计两周时间，建议进度安排如下：

选题，应该在上机实验之前完成

需求分析、概要设计可分配 4 学时完成

详细设计可分配 4 学时

调试和分析可分配 10 学时。

2 学时的机动，可用于辩论及按教师要求修改课程设计说明书。

注：只用课内上机时间一般不能完成设计任务，所以需要学生自行安排时间做补充。

六、推荐参考资料(不少于 3 篇)

[1] 苏仕华等编著，数据结构课程设计，机械工业出版社，2007

[2] 严蔚敏等编著，数据结构（C语言版），清华大学出版社，2003

[3] 严蔚敏等编著，数据结构题集（C语言版），清华大学出版社，2003

指导教师 _____ 签名日期 _____ 年 ____ 月 ____ 日

系主任 _____ 审核日期 _____ 年 ____ 月 ____ 日

摘 要

分析了 matlab ， mathmatic ， maple 等数学软件对一元多项式的计算过程，步骤后。由于这些软件比拟大功能齐全，但是实用性不强。因此，利用 microsoft visual studio 6.0 开发工具，编程实现了一元多项式的加法、减法、乘法的计算器系统，该系统具有一元多项式的加法、减法、乘法等功能。

关键词：一元多项式；软件；计算

目 录

1 需求分析	1
2 开发及运行平台.....	2
3 概要设计.....	3
4 详细设计.....	4
5 调试分析.....	8
6 测试结果.....	9
7 结论.....	11
致 谢.....	12
参考文献.....	13
附 录.....	16

1 需求分析

动态链表结构下的一元多项式的加法、减法、乘法的实现。

设有一元多项式 $A_m(x)$ 和 $B_n(x)$.

$$A_m(x) = A_0 + A_1x^1 + A_2x^2 + A_3x^3 + \dots + A_mx^m$$

$$B_n(x) = B_0 + B_1x^1 + B_2x^2 + B_3x^3 + \dots + B_nx^n$$

实现求 $M(x) = A_m(x) + B_n(x)$ 、 $M(x) = A_m(x) - B_n(x)$ 和 $M(x) = A_m(x) \times B_n(x)$ 。

(1) 输入形式和输入值范围：

输入的系数为 `float` 类型，输入的幂为 `int` 类型

请选择:1

请输入你要运算的第一个一元多项式的项数：

2

请输入第 1 项的系数和指数：

系数:1

指数:1

请输入第 2 项的系数和指数：

系数:1

指数:2

(2) 输出形式

请选择:5

一元多项式 A 为：

$$x + x^2$$

一元多项式 B 为：

$$4x^4 + 5x^5 + 6x^6$$

(3) 程序所能到达的功能

1) 首先判定多项式是否稀疏；

2) 采用动态存储结构实现；

3) 结果 $M(x)$ 中无重复阶项和无零系数项;

4) 要求输出结果的升幂和降幂两种排列情况

(4) 测试数据: 包括正确地输入及其输出结果和含有错误的输入及其输出结果。

正确的输入:

请选择:5

一元多项式 A 为:

$x+x^2$

一元多项式 B 为:

$4x^4+5x^5+6x^6$

错误的输入:

请输入第 1 项的系数和指数:

系数:1

指数:1

请输入第 2 项的系数和指数:

系数:2

指数:1

输入的该项指数与多项式中已存在的某项相同,请重新创立一个正确的多项式

请输入第 1 项的系数和指数:

2 开发及运行平台

Windows 7 alienware 平台上, Microsoft Visual studio 6.0 环境下开发。所以建议用户在使用时, 在 windows 7 平台上使用。

3 概要设计

数据类型的定义：

```
struct Node
{
float coef;//  结点类型，系数
int exp;//  指数
};

typedef Node polynomial;
```

```
struct LNode
{
polynomial data;//  链表类型
LNode *next;
};
```

```
typedef LNode* Link;
```

模块调用关系如图 1 所示：

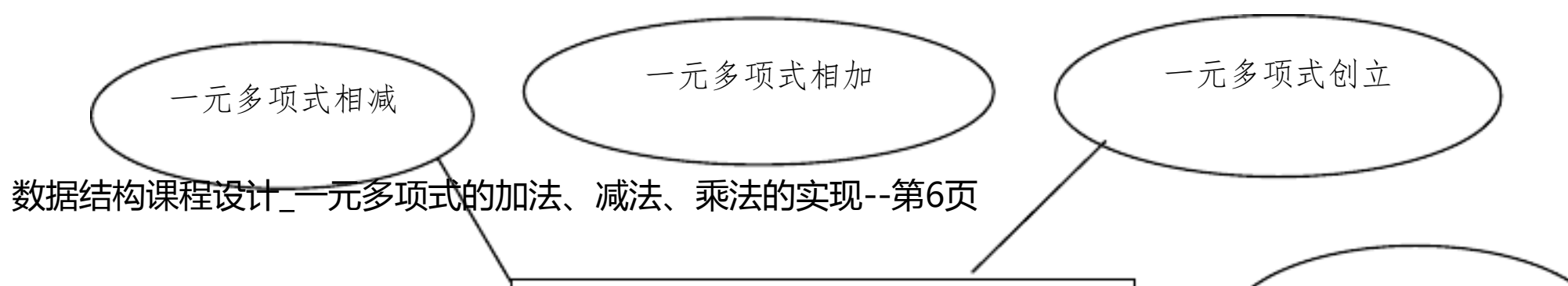




图 1一元多项式计算器模块调用图

4 详细设计

加法模块：

```

void PolyAdd(Link &pc,Link pa,Link pb)
{
Link p1,p2,p,pd;
CopyLink(p1,pa);
CopyLink(p2,pb);
pc=new LNode;
pc->next<-NULL;
p<-pc;
p1<-p1->next;
p2<-p2->next;
while(p1 不为空且 p2 不为空)
{if(p1->data.exp<p2->data.exp)
then : p->next=p1;p=p->next; p1=p1->next;
else if(p1->data.exp>p2->data.exp)
then : p->next<-p2; p<-p->next;p2<-p2->next;
else
{<-p1->data.coef+p2->data.coef;// 指数相同，系数相加
不等于 0)
{p->next<-p1;
p<-p->next;
p1<-p1->next;
p2<-p2->next;
}
else
{pd<-p1;
p1<-p1->next;
p2<-p2->next;
}
}
}

```



```

delete pd;//    如果系数为 0，那么删除该项
} }
}

if(p1!=NULL)
then : p->next=p1;endif
if(p2!=NULL)
then : p->next=p2;endif
}

```

减法模块，如图 2 所示：

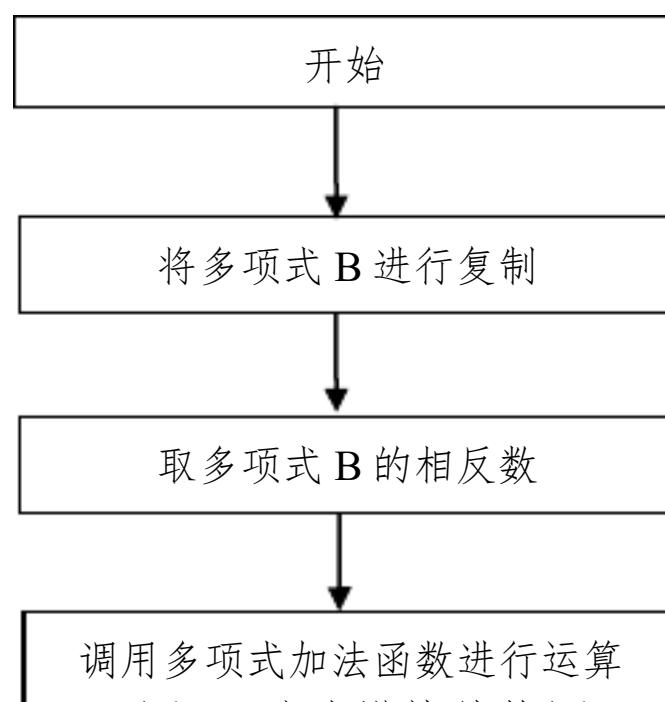


图 2 减法模块结构图

```

void PolySubstract(Link &pc,Link pa,Link pb)
{
Link p,pt;
CopyLink(pt,pb);
p<-pt;
while(p 不为空)
{ (p->data).coef<- -(p->data).coef;//    被减的多项式前加    号
p<- (p->next) ;
}
PolyAdd(pc,pa,pt);//    调用多项式加法运算函数

```

```
}
```

乘法模块：如图 3 所示

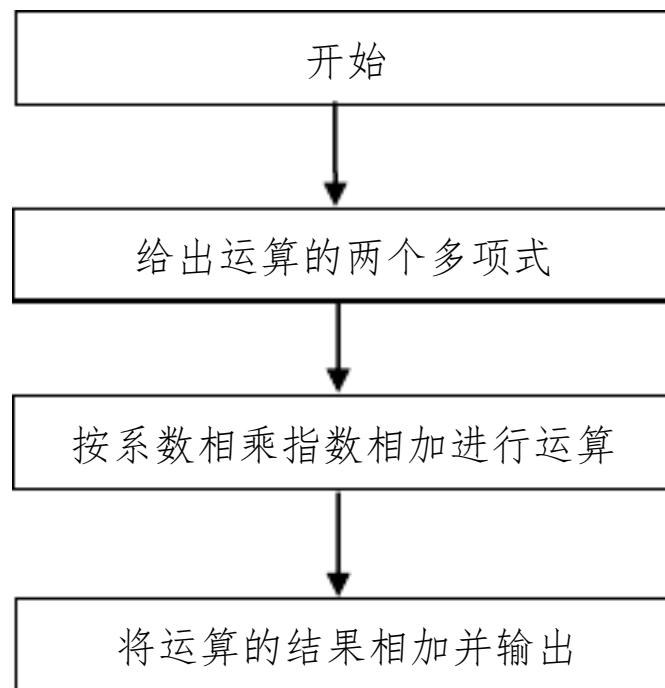


图 3 乘法模块结构图

```
void PolyMultiply(Link &pc,Link pa,Link pb)
```

```
{Link p1,p2,p,pd,newp,t;
```

```
pc=new LNode;
```

```
pc->next=NULL;
```

```
p1=pa->next;
```

```
p2=pb->next;
```

```
while(p1 不为空)
```

```
{ pd=new LNode;
```

```
pd->next=NULL;
```

```
p=new LNode;
```

```
p->next=NULL;
```

```
t=p;
```

```
while(p2)
```

```
{ newp=new LNode;
```

```
newp->next=NULL;
```

```
newp->data.coef=p1->data.coef*p2->data.coef;//
```

系数相乘

```

    t->next=newp;

    t=t->next;

    p2=p2->next;
}

PolyAdd(pd,pc,p);//    调用加法函数

CopyLink(pc,pd);

p1=p1->next;

p2=pb->next;

DestroyLink(p);

DestroyLink(pd);

}

}

```

调试分析

(1) 调试过程中遇到的问题：

独立测试各个模块的功能时发现在创立链表后和另一个进行运算后多余的存储单元没有释放而造成内存的泄漏，还有对于链表的运算时结束条件掌握不透彻导致没有按方案地去结束，比方在用 **For** 循环及 **While** 循环时没有正确地判断指针的移动与结束条件而得不到自己想要的结果。进入函数内部调试时发现有用没有初始化的变量，还有赘余的语句扰乱了代码的健壮性。在主函数中对各个函数的调用时没有一个清晰的思路，使程序显得很混乱，给调试造成了很大困难。对非法操作控制的不够完善，例如缺少对越界访问及其非法数据的控制机制，使程序的平安性下降。

2) 算法的时空分析

该程序的实现是用链表实现的，所以时间复杂度和空间复杂度主要来自于链表的操作，都是 $O(n)$ 。

(3) 经验和体会

编写的程序不但要拿来使用，还要给别人查看，以便代码的维护。所以代码编写的风格尽量标准，清晰。变量要尽量少定义，结构体采用简单的。另外，对指针的使用要小心，尽量在定义的时候就进行初始化，防止野指针，指针的使用涉及到内存的分配。

测试结果

判断稀疏，如图 4 所示：

```
请选择:1
请输入你要运算的第一个一元多项式的项数:
2
请输入第1项的系数和指数:
系数:
1
指数:1
请输入第2项的系数和指数:
系数:2
指数:2
该多项式是连续的哈
```

图 4 判断多项式是否稀疏

加法运算，如图 5 所示：

```
请选择:2
设相加的两个一元多项式为A和B则:
A的多项式为:  $x+2x^2$ 
B的多项式为:  $x+2x^2+3x^3$ 
相加后的结果为:  $2x+4x^2+3x^3$ 
```

5加法测试结果

减法运算，如图 6 所示：

```
请选择:3
设相减的两个一元多项式为A和B则:
A的多项式为:  $x+2x^2$ 
B的多项式为:  $x+2x^2+3x^3$ 
相减后的结果为:  $-3x^3$ 
```

图 6 减法测试结果

乘法运算，如图 7 所示：

```
请选择:4
设相乘的两个一元多项式为A和B则:
A的多项式为:  $x+2x^2$ 
B的多项式为:  $x+2x^2+3x^3$ 
相乘后的结果为:  $x^2+4x^3+7x^4+6x^5$ 
```

图 7 乘法测试结果

降幂升幂排序，如图 8 所示

```
退出
=====
请选择:7
降幂排序
 $3x^3+4x^2+2x$ 
请按任意键继续. . .
```

图 8 降幂升幂排序测试结果

结论

该程序根本实现了要求的顺序结构、动态链表结构下的一元多项式的加法、减法、乘法等功能。代码较为冗余，可读性较差，可以多添加一些提示语句以及注释。

在这次课程设计中我又进一步地了解了数据结构中算法的核心思想的重要性，懂得了一个程序地好坏关键在于算法是否优秀，一个好的优秀的算法可以使我们的程序更加完善，平安性更高以及有更高的效率。这次设计中我发现了自己的许多缺乏，如对指针的机制掌握的还不是很透彻，有的时候会出现指针指向错误以及空指针的错误，还有不能很好地分析自己算法地复杂度以及不能很好地使用控制机制使自己的程序流畅地运行。

谢

料，让我在编写代码的过程中少走了很多弯路。

参考文献

- [1] 苏仕华等编著. 数据结构课程设计. 机械工业出版社. 2007
- [2] 严蔚敏等编著. 数据结构 (C语言版). 清华大学出版社. 2003
- [3] 严蔚敏等编著. 数据结构题集 (C语言版). 清华大学出版社, 2003
- [
- [5]. 陈清华 朱红主编. Visual C++ 课程设计案例精选与编程指导. 东南大学出版社, 2003. 06, 南京
- [6]. 刘振安等编著. C++ 程序设计课程设计. 机械工业出版社, 2004. 08, 北京

附 录

附录 1 源程序清单

```
#include<iostream>// 标准输入输出流
#include<conio.h>// 使程序中可用键盘输入函数
#include<stdlib.h>// 使程序中可用系统标准输出函数
#include<math.h>// 调用数学库函数
using namespace std;// 命名空间 std 内定义的所有标识符均有效
struct Node{
float coef;// 结点类型, 系数
int exp;// 指数
};
typedef Node polynomial;
struct LNode{
polynomial data;// 链表类型
LNode *next;
};
typedef LNode* Link;
void CreateLink(Link &L,int n);
void PrintList(Link L);
void PolyAdd(Link &pc,Link pa,Link pb);
void PolySubtract(Link &pc,Link pa,Link pb);
```

```

void CopyLink(Link &pc,Link pa);
void PolyMultiply(Link &pc,Link pa,Link pb);
int JudgeIfExpSame(Link pa,Link e);
void DestroyLink(Link &L);
int CompareIfNum(int i);
void DestroyLink(Link &L)
{Link p;
p=L->next;
while(p)
{ L->next=p->next;
delete p;
p=L->next;
}
delete L;
L=NULL;
}

void CreateLink(Link &L,int n)
{if(L!=NULL)
{DestroyLink(L);}
Link p,newp;
L=new LNode;
L->next=NULL;
(L->data).exp=-1;// 创立头结点
p=L;
for(int i=1;i<=n;i++)
{ newp=new LNode;
        请输入第          项的系数和指数
        系数
cin>>(newp->data).coef;
        指数
cin>>(newp->data).exp;
if(newp->data.exp<0)
        您输入有误, 指数不允许为负值
delete newp;
}
}

```

```

i--;
continue;
}
newp->next=NULL;
p=L;
if(newp->data.coef==0)
    系数为零, 重新输入
delete newp;
i--;
continue;
}
while((p->next!=NULL)&&((p->next->data).exp<(newp->data).exp))
{ p=p->next; //p    指向指数最小的那一个 }
if(!JudgeIfExpSame( L, newp))
{newp->next=p->next;
p->next=newp;
}
else

```

输入的该项指数与多项式中已存在的某项相同,请重新创立一个正确的多项式

```

delete newp;
DestroyLink(L);
CreateLink(L,n); //    创立多项式没有成功, 递归调用重新创立
break;
}
}
JudgeIf(L);//    判断稀疏
}
void JudgeIf(Link L)
{Link p;
int flag=0;//    判断是否为稀疏的标志
p=L->next;
while(p->next!=NULL)
{if(abs(p->next->data.exp-p->data.exp)>1)

```

```

        flag=1;
    else
        flag=0;
    p=p->next;
}
if(flag)
    该多项式是稀疏的哈
else
    该多项式是连续的哈
}
int JudgeIfExpSame(Link L,Link e)
{Link p;
p=L->next;
while(p!=NULL&&(e->data.exp!=p->data.exp))
    p=p->next;
if(p==NULL)
return 0;
else return 1;
}
void PrintList(Link L)
{Link p;
if(L==NULL||L->next==NULL)
    该一元多项式为空!
else
{p=L->next;
if((p->data).coef>0)
{ if((p->data).exp==0)
    cout<<(p->data).coef;//    如果指数为 0 那么直接输出系数
else if((p->data).coef==1&&(p->data).exp==1)
    如果系数和指数均为 1, 那么输出 x
else if((p->data).coef==1&&(p->data).exp!=1)

else if((p->data).exp==1&&(p->data).coef!=1)

```

```

    }
    if((p->data).coef<0)
    {   if((p->data).exp==0)
        cout<<(p->data).coef;//      如果指数为 0，那么直接输出系数
    else if(p->data.coef==-1&&p->data.exp==1)
        如果系数为-1，指数为 1，那么输出-x
    else if(p->data.coef==-1&&p->data.exp!=1)

    else if(p->data.exp==1)
        如果指数为 1，那么输出系数倍 x

    }
    p=p->next;
    while(p!=NULL)
    {if((p->data).coef>0)//      系数大于 0 时输出情况
        {if((p->data).exp==0)

        else if((p->data).exp==1&&(p->data).coef!=1)

        else if((p->data).exp==1&&(p->data).coef==1)

        else if((p->data).coef==1&&(p->data).exp!=1)

        }
    if((p->data).coef<0)//      系数小于 0 时输出情况
    { if((p->data).exp==0)
        cout<<(p->data).coef;
    else if(p->data.coef==-1&&p->data.exp==1)

    else if(p->data.coef==-1&&p->data.exp!=1)

    else if(p->data.exp==1)

```

```

    }
    p=p->next;
    }
}
cout<<endl;
}
void CopyLink(Link &pc,Link pa)
{Link p,q,r;
pc=new LNode;
pc->next=NULL;
r=pc;
p=pa;
while(p->next!=NULL)
{ q=new LNode;
q->data.coef=p->next->data.coef;
q->data.exp=p->next->data.exp;
r->next=q;
q->next=NULL;
r=q;
p=p->next;}
}
void PolyAdd(Link &pc,Link pa,Link pb)
{ Link p1,p2,p,pd;
CopyLink(p1,pa);
CopyLink(p2,pb);
pc=new LNode;
pc->next=NULL;
p=pc;
p1=p1->next;
p2=p2->next;
while(p1!=NULL&& p2!=NULL)
{if(p1->data.exp<p2->data.exp)

```

```

    {
    p->next=p1;
    p=p->next;
    p1=p1->next;
    }
else if(p1->data.exp>p2->data.exp)
    {
    p->next=p2;
    p=p->next;
    p2=p2->next;
    }
else
{p1->data.coef=p1->data.coef+p2->data.coef;
if(p1->data.coef!=0)
    {
    p->next=p1;
    p=p->next;
    p1=p1->next;
    p2=p2->next;
    }
else
{pd=p1;
p1=p1->next;
p2=p2->next;
delete pd;//    如果系数为 0，那么删除该项
    }
}
}
if(p1!=NULL)
{ p->next=p1;}
if(p2!=NULL)
{ p->next=p2;}
}
void PolySubstract(Link &pc,Link pa,Link pb)

```

```

{Link p,pt;
CopyLink(pt,pb);
p=pt;
while(p!=NULL)
{ (p->data).coef=(-(p->data).coef);//      被减的多项式前加    号
  p=p->next;
}
PolyAdd(pc,pa,pt);//      调用多项式加法运算函数
DestroyLink(pt);
}
void Clear()

} //让用户重新选择
void PolyMultiply(Link &pc,Link pa,Link pb)
{Link p1,p2,p,pd,newp,t;
pc=new LNode;
pc->next=NULL;
p1=pa->next;
p2=pb->next;
while(p1!=NULL)
{pd=new LNode;
  pd->next=NULL;
  p=new LNode;
  p->next=NULL;
  t=p;
  while(p2)
  {newp=new LNode;
    newp->next=NULL;
    newp->data.coef=p1->data.coef*p2->data.coef;//      系数相乘
    newp->data.exp=p1->data.exp+p2->data.exp;//      指数相加
    t->next=newp;
    t=t->next;
    p2=p2->next;
  }
  p1=p1->next;
}
}

```



```

    }
    PolyAdd(pd,pc,p);
    CopyLink(pc,pd);
    p1=p1->next;
    p2=pb->next;
    DestroyLink(p);
    DestroyLink(pd);
}
}
void Menu()

cout<<endl;

```

一元多项式的加、减、乘运算

创立要运算的两个一元多项式
 将两个一元多项式相加
 将两个一元多项式相减
 将两个一元多项式相乘
 显示两个一元多项式
 销毁所创立的二个多项式
 退出

请选择

```

}
int CompareIfNum(int i)
{if(i>0&& i<8)
    return 0;
else return 1;//    返回 1 时出错，因为菜单中只有 1~7
}

void main()
{int n;
Link L,La=NULL,Lb=NULL;//La,Lb 分别为创立的两个多项式

```

```

while(1)
{ Menu(); // 调用菜单函数
  cin>>choose;
  switch(choose)
  { case 1:
      请输入您要运算的第一个一元多项式的项数
      cin>>n;
      if(CompareIfNum(n)==1)
          您的输入有误, 请重新输入……
      Clear();
      break;
    }
  CreateLink(La,n);
      请输入您要运算的第二个一元多项式的项数
  cin>>n;
  if(CompareIfNum(n)==1)
      您的输入有误, 请重新输入……
  Clear();
  break;
}
CreateLink(Lb,n);
Clear();
break;
case 2:
  if(La==NULL||Lb==NULL)
      您的多项式创立有误, 请重新选择……
  Clear();
  break;
}
PolyAdd(L,La,Lb);

```

设相加的两个一元多项式为 A 和 B 那么:

PrintList(La);

的多项式为:

PrintList(Lb);

相加后的结果为:

PrintList(L);

Clear();

DestroyLink(L);

break;

case 3:

if(La==NULL||Lb==NULL)

您的多项式创立有误, 请重新选择……

Clear();

break;

}

PolySubstract(L,La,Lb);

设相减的两个一元多项式为 A 和 B 那么:

的多项式为:

PrintList(La);

的多项式为:

PrintList(Lb);

相减后的结果为:

PrintList(L);

Clear();

DestroyLink(L);

break;

case 4:

您的多项式创立有误，请重新选择……

```
Clear();  
break;  
}
```

```
PolyMultiply(L,La,Lb);
```

设相乘的两个一元多项式为 A 和 B 那么：

的多项式为：

```
PrintList(La);
```

的多项式为：

```
PrintList(Lb);
```

相乘后的结果为：

```
PrintList(L);  
DestroyLink(L);
```

```
Clear();  
break;  
case 5:
```

```
if(La==NULL||Lb==NULL)
```

您的多项式创立有误，请重新选择……

```
Clear();  
break;  
}
```

一元多项式 A 为：

```
PrintList(La);
```

一元多项式 B 为：

```
PrintList(Lb);
```

```
Clear();  
break;
```

```

if(La&&Lb)
{DestroyLink(La);
DestroyLink(Lb);
        多项式销毁成功!
Clear();
}
else
        多项式不存在, 请重新选择
Clear();
}
break;
case 7:
exit(0); //exit(0)        强制终止程序, 返回状态码 0 表示正常结束
default:
        您的输入有误, 请重新选择操作……
Clear();
break;
}
}
}

```

附录 2

用户使用说明

双击该软件目录下的 文件即可, 或那么右击, 然后翻开即可正常使用。

施工组织设计

本施工组织设计是本着“一流的质量、一流的工期、科学管理”来进行编制的。编制时，我公司技术开展部、质检科以及工程部经过精心研究、合理组织、充分利用先进工艺，特制定本施工组织设计。

一、工程概况：

西夏建材城生活区 27#、30#住宅楼位于银川市新市区，橡胶厂对面。

本工程由宁夏燕宝房地产开发开发，银川市规划建筑设计院设计。

本工程耐火等级二级，屋面防水等级三级，地震防烈度为 8 度，设计使用年限 50 年。

本工程建筑面积：27#m²；30# m²。室内地坪± m 为准，总长 27#m； 30# m。总宽 27#m； 30# m。设计室外地坪至檐口高度 18.6 00m，呈长方形布置，东西向，三个单元。

本工程设计屋面为坡屋面防水采用防水涂料。外墙水泥砂浆抹面，外刷浅灰色墙漆。内墙面除卫生间 200×300 瓷砖，高到顶外，其余均水泥砂浆罩面，刮二遍腻子；楼梯间内墙采用 50 厚胶粉聚苯颗粒保温。地面除卫生间 200×200 防滑地砖，楼梯

厚细石砼 1:1 水泥砂浆压光外, 其余均采用 50 厚豆石砼毛地面。楼梯间单元门采用楼宇对讲门, 卧室门、卫生间门采用木门, 进户门采用保温防盗门。本工程窗均采用塑钢单框双玻窗, 开启窗均加纱扇。本工程设计为节能型住宅, 外墙均贴保温板。

本工程设计为砖混结构, 共六层。根底采用 C30 钢筋砼条形根底, 上砌 MU30 毛石根底, 砂浆采用 M10 水泥砂浆。一、二、三、四层墙体采用 M10 混合砂浆砌筑 MU15 多孔砖; 五层以上采用 M 混合砂浆砌筑 MU15 多孔砖。

本工程结构中使用主要材料: 钢材: I 级钢, II 级钢; 砼: 根底垫层 C10, 根底底板、地圈梁、根底构造柱均采用 C30, 其余均 C20。

本工程设计给水管采用 PPR 塑料管, 热熔连接; 排水管采用 UPVC 硬聚氯乙烯管, 粘接; 给水管道安装除立管及安装 IC 卡水表的管段明设计外, 其余均暗设。

本工程设计采暖为钢制高频焊翅片管散热器。

本工程设计照明电源采用 BV—铜芯线, 插座电源等采用 BV—4 铜芯线; 除客厅为吸顶灯外, 其余均采用座灯。

二、 施工部署及进度方案

1、 工期安排

本工程合同方案开工日期: 2004 年 8 月 21 日, 竣工日期: 2005 年 7 月 10 日, 合同工期 315 天。方案 2004 年 9 月 15 日前

完成根底工程，2004年12月30日完成主体结构工程，2005年6月20日完成装修工种，安装工程穿插进行，于2005年7月1日前完成。具体进度方案详见附图—1（施工进度方案）。

2、施工顺序

(1)根底工程

工程定位线〔验线〕→挖坑→钎探〔验坑〕→砂砾垫层的施工→根底砼垫层→刷环保沥青→根底放线〔预检〕→砼条形根底→刷环保沥青→毛石根底的砌筑→构造柱砼→地圈梁→地沟→回填工。

(2)结构工程

结构定位放线〔预检〕→构造柱钢筋绑扎、定位〔隐检〕→砖墙砌筑〔+50cm线找平、预检〕→柱梁、顶板支模〔预检〕→梁板钢筋绑扎〔隐检、开盘申请〕→砼浇筑→下一层结构定位放线→重复上述施工工序直至顶。

(3)内装修工程

门窗框安装→室内墙面抹灰→楼地面→门窗安装、油漆→五金安装、内部清理→通水通电、竣工。

(4)外装修工程

外装修工程遵循先上后下原那么，屋面工程〔包括烟道、透气孔、压顶、找平层〕结束后，进行大面积装饰，塑钢门窗在装修中逐步插入。

三、施工准备

1、 现场道路

本工程北靠北京西路，南临规划道路，交通较为方便。

场内道路采用级配砂石铺垫，压路机压。

2、 机械准备

(1)设 2 台搅拌机，2 台水泵。

(2)现场设钢筋切断机 1 台，调直机 1 台，电焊机 2 台，1 台对焊机。

(3)现场设木工锯，木工刨各 1 台。

(4)回填期间设打夯机 2 台。

(5)现场设塔吊 2 台。

3、 施工用电

施工用电已由建设单位引入现场；根据工程特点，设总配电箱 1 个，塔吊、搅抖站、搅拌机、切断机、调直机、对焊机、木工棚、楼层用电、生活区各配置配电箱 1 个；电源均采用三相五线制；各分支均采用钢管埋地；各种机械均设置接零、接地保护。具体配电箱位置详见总施工平面图。

3、 施工用水

施工用水采用深井水自来水，并砌筑一蓄水池进行蓄水。楼层用水采用钢管焊接给水管，每层留一出水口；给水管不置蓄水池内，由潜水泵进行送水。

4、 生活用水

生活用水采用自来水。

5、劳动力安排

(1)结构期间:

瓦工 40 人; 钢筋工 15 人; 木工 15 人; 放线工 2 人; 材料 1 人; 机工 4 人; 电工 2 人; 水暖工 2 人; 架子工 8 人; 电焊工 2 人; 壮工 20 人。

(2)装修期间

抹灰工 60 人; 木工 4 人; 油工 8 人; 电工 6 人; 水暖工 10 人。

四、主要施工方法

1、施工测量放线

(1)施工测量根本要求

A、西夏建材城生活区 17#、30#住宅楼定位依据: 西夏建材城生活区工程总体规划图, 北京路、规划道路永久性定位

B、根据工程特点及<建筑工程施工测量规程>DBI01—21—95, 4、3、2 条, 此工程设置精度等级为二级, 测角中误差 ± 12 , 边长相对误差 1/15000。

C、根据施工组织设计中进度控制测量工作进度, 明确对工程效劳, 对工程进度负责的工作目的。

(2)工程定位

A、根据工程特点, 平面布置和定位原那么, 设置一横一纵两条主控线即 27#楼:〔A〕轴线和〔1〕轴线; 30#楼:〔A〕轴线和〔1〕轴线。根据主轴线设置两条次轴线即 27#楼:

〔H〕轴线和〔27〕轴线；30#楼：〔H〕轴线和〔27〕轴线。

B、主、次控轴线定位时均布置引桩，引桩采用木桩，后砌一水泥砂浆砖墩；并将轴线标注在四周永久性建筑物或构造物上，施测完成后报建设单位、监理单位确认后另以妥善保护。

C、控轴线沿结构逐层弹在墙上，用以控制楼层定位。

D、水准点：建设单位给定准点，建筑物±.500m。

(3)根底测量

A、在开挖前，基坑根据平面布置，轴线控制桩为基准定出基坑长、宽度，作为拉小线的依据；根据结构要求，条基外侧 1100mm 为砂砾垫层边，考虑放坡，撒上白灰线，进行开挖。

B、在垫层上进行根底定位放线前，以建筑物平面控制线为准，校测建筑物轴线控制桩无误后，再用经纬仪以正倒镜挑直法直接投测各轴线。

C、标高由水准点引测至坑底。

(4)结构施工测量

A、首层放线验收后，主控轴一引至外墙立面上，作为以上各层主轴线竖身高以测的基准。

B、施工层放线时，应在结构平面上校投测轴线，闭合后再测设细部尺寸和边线。

C、标高竖向传递设置 3 个标高点，以其平均点引测水平

线折平时，尽量将水准仪安置在测点范围内中心位置，进行测设。

2、基坑开挖

本工程设计地基换工，夯填砂砾垫层 1100mm；根据此特点，采用机械大开挖，留 200mm 厚进行挖工、铲平。

开挖时，根据现场实际土质，按标准要求 1:0.33 放坡，反铲挖掘机挖土。开挖出的土，根据现场实际情况，尽量留足需用的好土，多余土方挖出，防止二次搬运。

人工开挖时，由技术员抄平好水平控制小木桩，用方铲铲平。

挖掘机挖土应该从上而下施工，禁止采用挖空底脚的操作方法。机械挖土，先发出信号，挖土的时候，挖掘机操作范围内，不许进行其他工作，装土的时候，任何人都不能停留在装土车上。

3、砌筑工程

(1)材料

砖：MU15 多孔砖，毛石根底采用 MU30 毛石。

砂浆：±0.00 以下采用 M10 水泥砂浆，一、二、三、四层采用 M10 混合砂浆，五层以上采用 M7.5 混合砂浆。

(2)砌筑要求

A、开工前由工长对所管辖班组下发技术交底。

B、砌筑前应提前浇水湿润砖块，水率保持在 10%—

15%。

C、砌筑采用满铺满挤“三一砌筑法”，要求灰浆饱满，灰缝8—12mm。

D、外墙转角处应同时砌筑，内外墙交接处必须留斜槎，槎子长度不小于墙体高度的2/3，槎子必须平直、通顺。

E、隔墙与墙不同时砌筑又不留成斜槎时可于墙中引出阳槎或在墙的灰缝中预埋拉结筋，每道不少于2根。

F、接槎时必须将外表清理干净，浇水湿润，填实砂浆，保持灰缝平直。

G、砖墙按图纸要求每50mm设置2 ϕ 6钢筋与构造柱拉结，具体要求见结构总说明。

H、施工时需留置临时洞口，其侧边离交接处的墙面不少于500mm顶部设边梁。

4、钢筋工程

(1)凡进场钢筋须具备材质证明，原材料须取样试验，经复试合格后方可使用。

(2)钢筋绑扎前应仔细对照图纸进行翻样，根据翻样配料，施工前由工长对所管辖班组下发技术交底，准备施工工具，做好施工的准备工作的。

(3)板中受力钢筋搭接，I级钢30d，II级钢40d，搭接位置：上部钢筋在跨中1/3范围内，下部钢筋在支座1/3范围内。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/047061043015006152>