

Homework 10.26

5.28 采用教科书5.6节中给出的m元多项式的表示方法，写一个求m元多项式中第一变元的偏导数的算法。

类C描述：

```
1  Status MPListPartial(MPList &L){
2      //对广义表存储结构的多元多项式求第一变元z的偏导
3      for(p = L->hp->tp; p && p->exp; t = p, p = p->tp){ //t为前驱
4          if(p->tag) //表
5              Partial_coef(p->hp, p->exp);
6          else p->coef *= p->exp; //原子
7          p->exp--;
8      }
9      pre->tp = NULL;
10     if(p) free(p); //删除常数项
11     return OK;
12 }
13
14 Status Partial_coef(MPList &L, int e){ //递归乘以e
15     for(p = L; p; p = p->tp){
16         if(!p->tag) p->coef *= e;
17         else Partial_coef(p->hp, e);
18     }
19     return OK;
20 }
```

5.30 试按表头、表尾的分析方法重写求广义表的深度的递归算法。

类C描述：

```
1  int GListDepth(GList L){
2      //扩展线性链表存储方式
3      if(!L) return 1; //空表
4      if(L->tag == ATOM) return 0; //原子
5      for(max = 0, pp = L->hp; pp; pp = pp->tp){
6          dep = GListDepth(pp); //求以pp为头指针的子表深度
7          if(dep > max) max = dep;
8      }
9      return max + 1; //非空表的深度是各元素的深度的最大值加1
10 }
```

