

DS Homework 3

注：请使用 A4 纸作答，写上姓名学号，并于下一次上课时提交。

1、请用类 C/C++伪代码，写出一个交换单链表中两个结点位置的算法（自定义变量类型）。

本题意为交换单链表中指针 p 所指结点和其后继结点，令指针 $head$ 为该链表的头指针，指针 p 为指向该链表中要交换结点位置的一个结点。若指针 p 所指的结点存在后继结点，看它是否是头结点，如果是，则交换后还要改变该链表的 $head$ ，若不是头结点，则直接交换。

参考实现如下：

```
A1. q = p->next;
A2. if q != null
    {
        if p = head
        {   s = head.next; head.next = p; p.next = s;   }
        else
        {   r = head;
            while (r.next != p)
                r = r.next;
            r.next = q; p.next = q.next; q.next = p;
        }
    }
    else printf('no next node');
A3. End
```

2、试编写算法，将一个用循环链表表示的稀疏多项式分解成两个多项式，使这两个多项式中各自仅含奇次项或偶次项，并要求利用原链表中的结点空间构成这两个链表。

基本思想是在遍历表的过程中，将结点分为两类而插入到两个链表中，具体实现算法如下：

struct PNode{ // 结点类型

```
    float coef;
    int exp;
    struct PNode *next;
```

```
} Pnode, *LinkList;
```

```
void Split(Linklist &plyn, LinkList &odd)
```

```
{
    // plyn 指向稀疏多项式的循环链表头结点，odd 为新产生的
    // 仅含奇次项链表的头指针，运算后 plyn 链表中仅含偶次项
    LinkList p, q, s;
    odd = (LinkList) malloc(sizeof(PNode));
    odd->next = odd; // 建立奇次项循环空链表
    q = plyn; p = plyn->next; s = odd;
    while (p!=plyn)
    {
        if (p->exp%2==0)
        { q = p; p = p->next; }
        else
        {
            q->next = p->next; // 从原表中删去奇次项结点
            p->next = s->next;
            s->next = p; s = s->next; // 插入至新表中
            p = q->next;
        }
    }
}
```