

P15. 2.9 (1)

```
Status A(LinkedList L) {
```

```
    if (L && L → next) {
```

```
        Q = L; L = L → next; P = L;
```

```
        while (P → next) P = P → next;
```

```
        P → next = Q; Q → next = NULL
```

```
    }
```

```
    return OK;
```

```
} //A
```

当此单链表结点大于等于2时,

将原链表的第一个结点变为最后一个结点,

其余原第 k 个结点变为第 $(k-1)$ 个结点。



```
12) void BB(LNode *s, LNode *q) {
```

```
    p = s;
```

```
    while (p → next != q) p = p → next
```

```
    p → next = s;
```

```
} // BB
```

```
void AA(LNode *Pa, LNode *Pb) {
```

```
    BB(Pa, Pb);
```

```
    BB(Pb, Pa);
```

```
} // AA
```

将此单循环链表变成两个单循环链表 L_1 和 L_2 ,
 L_1 和 L_2 分别为两个结点之间的结点所形成的
链表和以第2个结点为第1个结点,以第1个结点前的
结点为末结点的链表



```

2.12. Status ListCompare (SqList *La, SqList *Lb)
{
    P = La; q = Lb;
    While (P == q && P && q)
    {
        P = P->next;
        q = q->next;
    }
    if (!P && q)
        return -1; // 表示 A < B
    if (!P && !q)
        return 0; // 表示 A = B
    if (!q && P)
        return 1; // 表示 A > B
    if (compare(P, q) > 0) // A'首元大于B'首元
        return 1; // 表示 A > B
    if (compare(P, q) < 0) // A'首元小于B'首元
        return -1; // 表示 A < B
}

```



2.14

```
int LENGTH (LinkList L)
```

```
{ P = L → next; if (!P) return 0; n = 0;  
  while (P)
```

```
  { P = P → next;  
    n = n + 1;
```

```
  }  
  return n;
```

```
}
```



```
typedef struct  
{ int coef;  
  int exp;  
} ElemType * P] Poly
```

2.39 float ValuePolynomial (Poly * P, float X₀)

```
{ Poly * q;  
  sum = 0.0;  
  q = Get(P) // q 指向第一个元素
```

```
while (q)
```

```
{ while (q->exp. )
```

```
{ sum = (q->coef) * (q->coef) + sum;
```

```
  q++; exp--;
```

```
}
```

```
return sum;
```

```
}
```

