

/\*采用顺序表保存一组整数，完成以下功能：

- 1) 初始化创建包含3个数据元素的表；
- 2) 在表尾插入一条数据元素；
- 3) 在表头删除一条数据元素；
- 4) 输出。

\*/

```
typedef struct
{
    int *elem;
    int len;
    int listsize;
}SqList;
void Init(SqList *L, int n);
void InsertTail(SqList *L, int e);
void DeleteTail(SqList *L, int *e);
void Print(SqList L);

int _tmain(int argc, _TCHAR* argv[])
{
    SqList L;
    Init(&L, 3);
    Print(L);
    InsertTail(&L, 10);
    Print(L);
    int e;
    DeleteTail(&L, &e);
    Print(L);
    return 0;
}

void Init(SqList *L, int n)
{
    L->listsize = MAXSIZE;
    L->len = n;
    L->elem = new int[MAXSIZE];
    for(int i=0; i<L->len; i++)
        L->elem[i] = i;
}

void InsertTail(SqList *L, int e)
{
    if(L->len < L->listsize)
        L->elem[L->len++] = e;
}

void DeleteTail(SqList *L, int *e)
```

```

{
    if(L->len)
        *e = L->elem[--L->len];
}
void Print(SqList L)
{
    for(int i=0; i<L.len; i++)
        printf("%d\t", L.elem[i]);
}

```

/\*采用不带头结点的单链表保存一组整数序列，完成以下功能：1) 初始化创建包含3个数据元素的表；2) 表头插入数据元素；3) 表头删除数据元素；4) 输出；5) P指向表中的某一个结点，将S所指之后的结点整体移动到表头\*/

```

typedef struct lnode
{
    int data;
    struct lnode *next;
} LNode, *Link;

void Init(Link *L, int n);
void Print(Link L);
void InsertHead(Link *L, int e);
void DeleteHead(Link *L, int *e);
Link GetElemLink(Link L, int e);
void ChangeLink(Link *L, Link S);

int _tmain(int argc, _TCHAR* argv[])
{
    Link L;
    int e;

    Init(&L, 6);
    InsertHead(&L, 10);
    Print(L);
    DeleteHead(&L, &e);
}

```

```

    Print(L);

    Link p = GetElemLink(L, 5);
    printf("%d\n", p->data);

    ChangeLink(&L, p);
    Print(L);
    return 0;
}

void InsertHead(Link *L, int e)
{
    Link s = new LNode;
    s->data = e;

    s->next = *L;
    *L = s;
}

void DeleteHead(Link *L, int *e)
{
    if(*L)
    {
        *e = (*L)->data;

        Link q = *L;
        *L = (*L)->next;

        delete q;
    }
}

Link GetElemLink(Link L, int e)

```

```

{
    Link p = L;
    while(p)
    {
        if(p->data == e)
            break;
        p = p->next;
    }
    return p;
}

void Init(Link *L, int n)
{
    *L = NULL; //空表的构造

    Link s;
    for(int i=0; i<n;i++)
    {
        s = new LNode;
        s->data = i;

        s->next = *L;
        *L = s;
    }
}

void Print(Link L)
{
    Link p = L;
    while(p)
    {
        printf("%d ", p->data);
        p = p->next;
    }
}

```

```

    }
    printf("\n");
}

void ChangeLink(Link *L, Link S)
{
    Link p = *L;
    if(p == S)
        return;
    while(p)
    {
        if(p->next == S)
            break;
        p = p->next;
    }
    p->next = NULL;

    p=S;
    while(p)
    {
        if(!p->next)
            break;
        p = p->next;
    }
    p->next= *L;
    *L = S;
}

```