

## 实验二参考答案

### 一、单选题

1、B 2、D 3、C 4、B 5、D 6.C 7. A

### 二、填空题

1、初始化 INITLATE (L)、求表长 LENGTH (L)、读表长 GET (L, i)、定位 LOCATE、(L, X)、插入 INSERT (L, X, i)、删除 DELETE (L, i)

2、 $O(n)$ 、 $O(1)$

3、前驱、前驱、后继、后继

4、 $p \rightarrow next$ 、 $a[p].next$

5、表头

6、前驱、后继

7、表尾、表头

8、 $a[j].next=a[i].next$ ; $a[i].next=j$ ;

```
***** new key
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <malloc.h>
```

```
#define LIST_INIT_SIZE 100
```

```
#define LISTINCREMENT 10
```

```
typedef char ElemType;
```

```
typedef struct {
```

```
    ElemType *elem;
```

```
    int length;
```

```
    int listsize;
```

```
    int incrementsize;
```

```
} SqList;
```

```
SqList InitList_Sq(SqList L) {
```

```
    L.elem=(ElemType*)malloc(sizeof(ElemType)*LIST_INIT_SIZE);
```

```
    L.length=0;
```

```
    L.listsize=LIST_INIT_SIZE;
```

```
    L.listsize=LIST_INIT_SIZE;
```

```
    L.incrementsize=LISTINCREMENT;
```

```
    return L; //增加了个返回，使其能完成初始化
```

```
}
```

```
int compare(SqList A,SqList B) {
```

```
    /* if A<B return -1;if A=B return 0;if A>B return 1 */
```

```
    int j;
```

```
    j=0;
```

```
    while(j<A.length && j<B.length){
```

```
        if(A.elem[j]<B.elem[j]) return(-1);
```

```

        else if(A.elem[j]>B.elem[j]) return(1);
            else j++;
    }
    if (A.length==B.length) return(0);
    else if(A.length<B.length) return(-1);
        else return(1);
}

void main() {
    SqList A,B;
    int result;
    A=InitList_Sq(A); //使 A 完成初始化
    B=InitList_Sq(B); //使 B 完成初始化
    printf("\n\n");
    printf("input value of List A:");
    scanf("%s",A.elem); A.length=strlen(A.elem);
    printf("\n\n"); printf("input value of List B:");
    scanf("%s",B.elem);
    B.length=strlen(B.elem);
    printf("\n\n");
    result=compare(A,B);
    if(result==1) printf("the result is:(A>B)");
    else if(result==-1) printf("the result is:(A<B)");
        else printf("the result is:(A=B)");
}

***** end new key

```

## 附 2：原程序

```

#define NULL 0
#define TRUE 1
#define FALSE 0

typedef char ElemType;

struct Node;
typedef struct Node *LNode,*LinkList;

struct Node{
    ElemType    data;
    struct Node *next;
};

```

```
LinkList CreateList_L(LinkList L, ElemType A[],int n)
```

```
{  
    /**/  
    int i;  
    L=NULL;  
  
    for(i=n-1;i>=0;i--){  
        LNode s=(LNode)malloc(sizeof(struct Node));  
        s->data=A[i];  
        s->next=L;  
        L=s;  
    }  
    return L;  
}
```

```
main()
```

```
{  
    ElemType A[5]={'a','b','c','d','e'};  
    int n;  
    LinkList L;  
    LNode p;  
  
    n=5;  
  
    p=CreateList_L(L,A,n);  
  
    printf("\n\n");  
    while(p!=NULL){  
        printf("%c\n",p->data);  
        p=p->next;  
    }  
}
```

□