

```

1  /*****
2  *                               《软件技术基础》上机实践 （三）
3  *                               ****
4  *   作业名称：Alpha 问题的链接表实现
5  *
6  *   作业描述：
7  *       从键盘接受用户输入若干个自然数，创建一个有序线性表。该有序表可以
8  *       为非减序，也可以为非增序。
9  *       其中，线性表的抽象数据类型的链接存储实现已经在 linear_list_lnk.c
10 *       中给出了部分的实现，并且插入排序、二分查找、输出有序表这些建立在线性
11 *       表的抽象数据类型之上的运算代码也已在文件 list_app.c 中实现。
12 *
13 *   作业要求：
14 *       根据文件 linear_list.h 中声明的线性表的抽象数据类型，以及linear_list.h
15 *       声明的接口，实现对链接表的基本操作，包括指定位置元素的获取，插入和删
16 *       除，即完成文件 linear_list_lnk.c。
17 *
18 *   提交说明：
19 *       提交材料：完整的 linear_list_lnk.c 源代码（包括注释）；
20 *
21 * ****
22 *
23 *   Copyright 2010-2013 by HAO Jiasheng <haojiasheng@gmail.com>
24 *
25 * **** */
26 #include <stdio.h>
27 #include <assert.h>
28 #include "linear_list.h"
29 #include "list_app.h"
30
31 #define EOL -1          /* End Of List, 输入结束标志 */
32
33
34 void do_purge(LIST list)
35 {
36     /* 删除重复出现的元素 */
37     printf("\n>> 删除重复出现的元素...\n");
38     purge_list(list);
39     /* 输出线性表 */
40     display_list(list);
41 }
42
43 void do_find(LIST list)
44 {
45     int x, i;
46
47     /*采用二分查找元素*/
48
49     printf ("\n>> 输入采用二分查找的元素: ");
50     scanf ("%d",&x);
51
52     i = search_list(list,x);
53     /* 输出结果 */
54     if ( i != E_NODE_NOT_FOUND )
55         printf("Found at the position: %d\n", i);
56     else
57         printf("Not found.\n");
58 }
59
60
61 void do_delete(LIST list)
62 {
63     int x, i;
64
65     /* 获取用户输入，并将线性表中对应的结点删除 */
66     printf("\n>> 输入要删除的元素: ");
67     scanf ("%d", &x);
68
69     /* 先定为值为 x 的结点，得到其位置，然后根据该位置执行删除操作 */
70     i = search_list(list, x);
71     if (E_SUCCESS != list_delete(list, i))
72         printf("Delete failed: node not found.\n");

```

```

73     /* 输出线性表 */
74     display_list(list);
75 }
76
77
78 /*****
79  * 主函数
80  *****/
81
82 /*
83  * 功能描述: 主函数, 获取用户输入, 并作相应处理, 打印输出
84  * 输入: 系统默认
85  * 输出: 0
86  */
87
88 int main(int argc, char *argv[])
89 {
90     LIST list;
91     int x;
92     int i;
93
94
95     /* 创建一个空的线性表 */
96     list = create_list();
97     assert(NULL != list);
98
99     /* 获取用户输入, 并插入到线性表中适当的位置 */
100    printf("\n>> 输入若干个自然数, 以 %d 结束输入: \n", EOL);
101
102    i = 0;
103    while (1) {
104        printf(" x[%d] = ", i);
105        scanf("%d", &x);
106        if (EOL == x)
107            break;
108
109        list_insert(list, list_length(list), x);
110        /* 对线性表进行排序 */
111        sort_list(list);
112        /* 输出线性表 */
113        display_list(list);
114
115        i++;
116    }
117
118    while (1) {
119        printf("\n>> [0-> 退出; 1-> 输出线性表; 2-> 删除重复结点; 3-> 查找结点; 4->
删除结点]: ");
120        scanf("%d", &x);
121
122        if (x == 0) break;
123
124        switch (x)
125        {
126            case 1:
127                display_list(list);
128                break;
129            case 2:
130                do_purge(list);
131                break;
132            case 3:
133                do_find(list);
134                break;
135            case 4:
136                do_delete(list);
137                break;
138        }
139    }
140
141    return 0;
142 }
143

```

```
144  /* Ends */  
145  
146
```