

```

1  /*****
2  *                               《软件技术基础》上机实践 （二）
3  *                               ****
4  *   作业名称：Alpha 问题的抽象数据类型实现
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，并可以对线
8  *       性表进行操作。该有序表可以为非减序，也可以为非增序。
9  *       线性表的抽象数据类型的接口定义在 linear_list.h 中已给出，同时基
10 *       于这些接口的线性表抽象运算（如插入排序、二分查找、输出有序表等）也
11 *       已在文件 list_app.c 中实现。
12 *       线性表的这个抽象数据类型的具体实现方式可以采用顺序表或者链接表，
13 *       分别在源文件 linear_list_seq.c 和 linear_list_lnk.c 中，其中已提供部
14 *       分操作实现作为参考。
15 *
16 *   作业要求：
17 *       根据文件 linear_list.h 中声明的线性表的抽象数据类型，完成线性表
18 *       抽象数据类型的两种不同具体实现。
19 *       （1）顺序表实现：在 linear_list_seq.c 实现顺序表的基本操作，包括
20 *       指定位置元素的获取、定位、插入和删除，即完成文件 linear_list_seq.c。
21 *       （2）链接表实现：在 linear_list_lnk.c 实现链接表的基本操作，包括
22 *       指定位置元素的获取、定位、插入和删除，即完成文件 linear_list_lnk.c。
23 *
24 *   提交说明：
25 *       提交材料：
26 *       （1）对整个程序中抽象数据类型的分析；
27 *       （2）完整的 linear_list_seq.c 源代码（包括注释）；
28 *       （3）完整的 linear_list_lnk.c 源代码（包括注释）。
29 *
30 ****
31 *
32 *   Copyright 2010-2016 by HAO Jiasheng <haojiasheng@gmail.com>
33 *
34 ****/
35 #include <stdio.h>
36 #include <assert.h>
37 #include "linear_list.h"
38 #include "list_app.h"
39
40 #define EOL -1          /* End Of List, 输入结束标志 */
41
42
43 void do_purge(LIST list)
44 {
45     /* 删除重复出现的元素 */
46     printf("\n>> 删除重复出现的元素...\n");
47     purge_list(list);
48     /* 输出线性表 */
49     display_list(list);
50 }
51
52 void do_find(LIST list)
53 {
54     int x, i;
55
56     /*采用二分查找元素*/
57
58     printf ("\n>> 输入采用二分查找的元素: ");
59     scanf ("%d",&x);
60
61     i = search_list(list,x);
62     /* 输出结果 */
63     if ( i != E_NODE_NOT_FOUND )
64         printf("Found at the position: %d\n", i);
65     else
66         printf("Not found.\n");
67 }
68
69
70 void do_delete(LIST list)
71 {
72     int x, i;

```

```

73
74     /* 获取用户输入，并将线性表中对应的结点删除 */
75     printf("\n>> 输入要删除的元素: ");
76     scanf("%d", &x);
77
78     /* 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 */
79     i = search_list(list, x);
80     if (E_SUCCESS != list_delete(list, i))
81         printf("Delete failed: node not found.\n");
82     /* 输出线性表 */
83     display_list(list);
84 }
85
86
87 /*****
88  * 主函数
89  *****/
90
91 /*
92  * 功能描述: 主函数，获取用户输入，并作相应处理，打印输出
93  * 输入: 系统默认
94  * 输出: 0
95  */
96
97 int main(int argc, char *argv[])
98 {
99     LIST list;
100     int x;
101     int i;
102
103     /* 创建一个空的线性表 */
104     list = create_list();
105     assert(NULL != list);
106
107     /* 获取用户输入，并插入到线性表中适当的位置 */
108     printf("\n>> 输入若干个自然数，以 %d 结束输入: \n", EOL);
109
110     i = 0;
111     while (1) {
112         printf(" x[%d] = ", i);
113         scanf("%d", &x);
114         if (EOL == x)
115             break;
116
117         list_insert(list, list_length(list), x);
118         /* 对线性表进行排序 */
119         sort_list(list);
120         /* 输出线性表 */
121         display_list(list);
122
123         i++;
124     }
125
126     while (1) {
127         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]: ");
128         scanf("%d", &x);
129
130         if (x == 0) break;
131
132         switch (x)
133         {
134             case 1:
135                 display_list(list);
136                 break;
137             case 2:
138                 do_purge(list);
139                 break;
140             case 3:
141                 do_find(list);
142                 break;
143

```

```
144         case 4:
145             do_delete(list);
146             break;
147         }
148     }
149
150     return 0;
151 }
152
153 /* Ends */
154
155
```