

```

1  /* SListTest.c */
2  #include <stdio.h>
3  #include <string.h>
4  #include <malloc.h>
5  #include <stdlib.h>
6  #include "SList.h"
7
8  static int IntCmp(const void *keyAddr, const void *dataAddr)
9  {
10     int *p1 = (int *)keyAddr;
11     int *p2 = (int *)dataAddr;
12     return (*p1 - *p2);
13 }
14 static void IntTraverse(void *keyAddr, void *outData)
15 {
16     int *p = (int *)keyAddr;
17     printf("%d\n", *p);
18 }
19
20 static int StringCmp(const void *keyAddr, const void *dataAddr)
21 {
22     char *p1 = *(char **)keyAddr;
23     char *p2 = *(char **)dataAddr;
24     return strcmp(p1, p2);
25 }
26 static void StringTraverse(void *keyAddr, void *outData)
27 {
28     char *p = *(char **)keyAddr;
29     printf("%s\n", p);
30 }
31 static void StringFree(void *keyAddr)
32 {
33     char *p = *(char **)keyAddr;
34     free(p);
35 }
36
37 int main()
38 {
39     LIST intList;
40     ListNew(&intList, sizeof(int), IntCmp, NULL);
41     int i = 0;
42     for (; i < 10; i++)
43     {
44         ListInsert(&intList, &i, LIST_FORWARD);
45     }
46     if (!ListEmpty(&intList))
47     {
48         printf("intList size is %d\n", ListSize(&intList));
49         ListTraverse(&intList, IntTraverse, NULL);
50     }
51     int intRemove = 3;
52     if (0 == ListRemove(&intList, &intRemove))
53     {
54         printf("intList remove %d success\n", intRemove);
55     }
56     else
57     {
58         printf("intList remove %d fail\n", intRemove);
59     }
60     if (0 == ListRemove(&intList, &intRemove))
61     {
62         printf("intList remove %d success\n", intRemove);
63     }
64     else
65     {
66         printf("intList remove %d fail\n", intRemove);
67     }
68     if (!ListEmpty(&intList))
69     {
70         printf("intList size is %d\n", ListSize(&intList));
71         ListTraverse(&intList, IntTraverse, NULL);
72     }
73     int intSearch = 2;

```

```

74     if (NULL != ListSearch(&intList, &intSearch))
75     {
76         printf("data %d is in intList\n", intSearch);
77     }
78     else
79     {
80         printf("data %d is not in intList\n", intSearch);
81     }
82     intSearch = 11;
83     if (NULL != ListSearch(&intList, &intSearch))
84     {
85         printf("data %d is in intList\n", intSearch);
86     }
87     else
88     {
89         printf("data %d is not in intList\n", intSearch);
90     }
91     ListDispose(&intList);
92
93     printf("\n\n\n");
94
95     LIST stringList;
96     ListNew(&stringList, sizeof(char *), StringCmp, StringFree);
97     char *name1 = strdup("pc");
98     char *name2 = strdup("pcwl513");
99     char *name3 = strdup("pcpc");
100    char *name4 = strdup("jerry");
101    char *name5 = strdup("jerry.peng");
102    char *name6 = strdup("yanglupu");
103    char *name7 = strdup("zhanglei");
104    char *name8 = strdup("lishanke");
105    ListInsert(&stringList, &name1, !LIST_FORWARD);
106    ListInsert(&stringList, &name2, !LIST_FORWARD);
107    ListInsert(&stringList, &name3, !LIST_FORWARD);
108    ListInsert(&stringList, &name4, !LIST_FORWARD);
109    ListInsert(&stringList, &name5, !LIST_FORWARD);
110    ListInsert(&stringList, &name6, !LIST_FORWARD);
111    ListInsert(&stringList, &name7, !LIST_FORWARD);
112    ListInsert(&stringList, &name8, !LIST_FORWARD);
113    if (!ListEmpty(&stringList))
114    {
115        printf("stringList size is %d\n", ListSize(&stringList));
116        ListTraverse(&stringList, StringTraverse, NULL);
117    }
118    char *strRemove = "pcpc";
119    if (0 == ListRemove(&stringList, &strRemove))
120    {
121        printf("stringList remove %s success\n", strRemove);
122    }
123    else
124    {
125        printf("stringList remove %s fail\n", strRemove);
126    }
127    if (0 == ListRemove(&stringList, &strRemove))
128    {
129        printf("stringList remove %s success\n", strRemove);
130    }
131    else
132    {
133        printf("stringList remove %s fail\n", strRemove);
134    }
135    if (!ListEmpty(&stringList))
136    {
137        printf("stringList size is %d\n", ListSize(&stringList));
138        ListTraverse(&stringList, StringTraverse, NULL);
139    }
140    char *strSearch = "yanglupu";
141    if (NULL != ListSearch(&stringList, &strSearch))
142    {
143        printf("data %s is in stringList\n", strSearch);
144    }
145    else
146    {

```

```
147         printf("data %s is not in stringList\n", strSearch);
148     }
149     strSearch = "123";
150     if (NULL != ListSearch(&stringList, &strSearch))
151     {
152         printf("data %s is in stringList\n", strSearch);
153     }
154     else
155     {
156         printf("data %s is not in stringList\n", strSearch);
157     }
158     ListDispose(&stringList);
159     return 0;
160 }
```