```c
/* DListTest.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include "DList.h"

static int IntCmp(const void *keyAddr, const void *dataAddr)
{
    int *p1 = (int *)keyAddr;
    int *p2 = (int *)dataAddr;
    return (*p1 - *p2);
}
static void IntTraverse(void *keyAddr, void *outData)
{
    int *p = (int *)keyAddr;
    printf("%d\n", *p);
}

static int StringCmp(const void *keyAddr, const void *dataAddr)
{
    char *p1 = *(char **)keyAddr;
    char *p2 = *(char **)dataAddr;
    return strcmp(p1, p2);
}
static void StringFree(void *keyAddr)
{
    char *p = *(char **)keyAddr;
    free(p);
}
static void StringTraverse(void *keyAddr, void *outData)
{
    char *p = *(char **)keyAddr;
    printf("%s\n", p);
}

int main()
{
    LIST intList;
    ListNew(&intList, sizeof(int), IntCmp, NULL);
    int i = 0;
    for (; i < 10; i++)
    {
        ListInsert(&intList, &i, LIST_FORWARD);
    }
    if (!ListEmpty(&intList))
    {
        printf("intList size is %d\n", ListSize(&intList));
        ListTraverse(&intList, IntTraverse, NULL);
    }
    int intRemove = 3;
    if (0 == ListRemove(&intList, &intRemove))
    {
        printf("intList remove %d success\n", intRemove);
    }
    else
    {
        printf("intList remove %d fail\n", intRemove);
    }
    if (0 == ListRemove(&intList, &intRemove))
    {
        printf("intList remove %d success\n", intRemove);
    }
    else
    {
        printf("intList remove %d fail\n", intRemove);
    }
    if (!ListEmpty(&intList))
    {
        printf("intList size is %d\n", ListSize(&intList));
        ListTraverse(&intList, IntTraverse, NULL);
    }
    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     }
```