

```

1  /* LinkStackTest.c */
2  #include <stdio.h>
3  #include <string.h>
4  #include <stdlib.h>
5  #include "LinkStack.h"
6
7  static void StringFree(void *elemAddr)
8  {
9      free(*(char **)elemAddr);
10 }
11
12 int main(void)
13 {
14     STACK intStack;
15     StackNew(&intStack, sizeof(int), NULL);
16     int i = 0;
17     for (; i < 10; i++)
18     {
19         if (0 == StackPush(&intStack, &i))
20         {
21             printf("intStack push key %d success\n", i);
22         }
23         else
24         {
25             printf("intStack push key %d fail\n", i);
26         }
27         if (!StackEmpty(&intStack))
28         {
29             int intTop;
30             if (0 == StackTop(&intStack, &intTop))
31             {
32                 printf("top of intStack is %d\n", intTop);
33             }
34             else
35             {
36                 printf("get top element of intStack fail\n");
37             }
38         }
39     }
40     if (!StackEmpty(&intStack))
41     {
42         printf("size of intStack is %d\n", StackSize(&intStack));
43     }
44
45     for(; i >= 0; i--)
46     {
47         int intPop;
48         if (0 == StackPop(&intStack, &intPop))
49         {
50             printf("data %d pop from intStack\n", intPop);
51         }
52         else
53         {
54             printf("pop from intStack fail\n");
55         }
56     }
57
58     StackDispose(&intStack);
59
60     printf("\n\n\n");
61
62     STACK stringStack;
63     StackNew(&stringStack, sizeof(char *), StringFree);
64     char *name1 = strdup("jerry");
65     char *name2 = strdup("pc");
66     char *name3 = strdup("pcwl513");
67     char *name4 = strdup("pcpc");
68
69     char *strTop = NULL;
70     StackPush(&stringStack, &name1);
71     if (0 == StackTop(&stringStack, &strTop))
72     {
73         printf("top of stringStack is %s\n", strTop);

```

```

74     }
75     else
76     {
77         printf("get top element of stringStack fail\n");
78     }
79     StackPush(&stringStack, &name2);
80     StackPush(&stringStack, &name3);
81     StackPush(&stringStack, &name4);
82     if (0 == StackTop(&stringStack, &strTop))
83     {
84         printf("top of stringStack is %s\n", strTop);
85     }
86     else
87     {
88         printf("get top element of stringStack fail\n");
89     }
90     if (!StackEmpty(&stringStack))
91     {
92         printf("size of stringStack is %d\n", StackSize(&stringStack));
93     }
94
95     i = 5;
96     for(; i >= 0; i--)
97     {
98         char *strPop = NULL;
99         if (0 == StackPop(&stringStack, &strPop))
100         {
101             printf("data %s pop from stringStack\n", strPop);
102         }
103         else
104         {
105             printf("pop from stringStack fail\n");
106         }
107     }
108
109     StackDispose(&stringStack);
110     return 0;
111 }

```