

```

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

```

```

74     if (0 <= pos)
75     {
76         printf("the position of data %d in intVector is %d\n", intSearch, pos);
77     }
78     else
79     {
80         printf("data %d is not in intVector\n", intSearch);
81     }
82     intSearch = 11;
83     pos = VectorSearch(&intVector, &intSearch, 1);
84     if (0 <= pos)
85     {
86         printf("the position of data %d in intVector is %d\n", intSearch, pos);
87     }
88     else
89     {
90         printf("data %d is not in intVector\n", intSearch);
91     }
92     VectorMakeEmpty(&intVector);
93     if (VectorEmpty(&intVector))
94     {
95         printf("intVector is made empty success\n");
96     }
97     else
98     {
99         printf("intVector is made empty fail\n");
100    }
101
102    for (i = 20; i > 10; i --)
103    {
104        VectorInsert(&intVector, &i);
105    }
106    if (!VectorEmpty(&intVector))
107    {
108        printf("intVector size = %d\n", VectorSize(&intVector));
109        VectorTraverse(&intVector, IntTraverse, NULL);
110    }
111    VectorDispose(&intVector);
112
113    printf("\n\n");
114
115    VECTOR stringVector;
116    VectorNew(&stringVector, sizeof(char *), 8, 0, StringCmp, StringFree);
117    char *name1 = strdup("jerry");
118    char *name2 = strdup("pc");
119    char *name3 = strdup("pcwl513");
120    char *name4 = strdup("pcpc");
121    char *name5 = strdup("zhanglei");
122    char *name6 = strdup("lishanke");
123    char *name7 = strdup("yanglupu");
124    char *name8 = strdup("jerry.peng");
125    VectorInsert(&stringVector, &name1);
126    VectorInsert(&stringVector, &name2);
127    VectorInsert(&stringVector, &name3);
128    VectorInsert(&stringVector, &name4);
129    VectorInsert(&stringVector, &name5);
130    VectorInsert(&stringVector, &name6);
131    VectorInsert(&stringVector, &name7);
132    VectorInsert(&stringVector, &name8);
133    if (!VectorEmpty(&stringVector))
134    {
135        printf("stringVector size = %d\n", VectorSize(&stringVector));
136        VectorTraverse(&stringVector, StringTraverse, NULL);
137    }
138    char *strRemove = "jerry.peng";
139    if (0 == VectorRemove(&stringVector, &strRemove))
140    {
141        printf("stringVector remove %s success\n", strRemove);
142    }
143    else
144    {
145        printf("stringVector remove %s fail\n", strRemove);
146    }

```

```
147     if (0 == VectorRemove(&stringVector, &strRemove))
148     {
149         printf("stringVector remove %s success\n", strRemove);
150     }
151     else
152     {
153         printf("stringVector remove %s fail\n", strRemove);
154     }
155     if (!VectorEmpty(&stringVector))
156     {
157         printf("stringVector size = %d\n", VectorSize(&stringVector));
158         VectorTraverse(&stringVector, StringTraverse, NULL);
159     }
160     char *strSearch = "zhanglei";
161     pos = VectorSearch(&stringVector, &strSearch, 1);
162     if (0 <= pos)
163     {
164         printf("the position of data %s in stringVector is %d\n", strSearch, pos);
165     }
166     else
167     {
168         printf("data %s is not in stringVector\n", strSearch);
169     }
170     strSearch = "123";
171     pos = VectorSearch(&stringVector, &strSearch, 1);
172     if (0 <= pos)
173     {
174         printf("the position of data %s in stringVector is %d\n", strSearch, pos);
175     }
176     else
177     {
178         printf("data %s is not in stringVector\n", strSearch);
179     }
180     VectorDispose(&stringVector);
181     return 0;
182 }
```