

```

1  /* AvlTest.c */
2  #include <stdio.h>
3  #include <string.h>
4  #include <malloc.h>
5  #include <stdlib.h>
6  #include "Avl.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
15 static void IntTraverse(void *dataAddr)
16 {
17     int *p = (int *)dataAddr;
18     printf("%d\n", *p);
19 }
20
21 static int StringCmp(const void *keyAddr, const void *dataAddr)
22 {
23     char *p1 = *(char **)keyAddr;
24     char *p2 = *(char **)dataAddr;
25     return strcmp(p1, p2);
26 }
27
28 static void StringTraverse(void *dataAddr)
29 {
30     char *p = *(char **)dataAddr;
31     printf("%s\n", p);
32 }
33
34 static void StringFree(void *dataAddr)
35 {
36     char *p = *(char **)dataAddr;
37     free(p);
38 }
39
40 int main()
41 {
42     AVLTree intAvl;
43     AvlNew(&intAvl, sizeof(int), IntCmp, NULL);
44
45     int i = 0;
46     for (; i < 10; i++)
47     {
48         AvlInsert(&intAvl, &i);
49     }
50     if (!AvlEmpty(&intAvl))
51     {
52         printf("intAvl size is %d\n", AvlSize(&intAvl));
53         printf("intAvl height is %d\n", AvlHeight(&intAvl));
54         printf("intAvl traverse pre:\n");
55         AvlTravPre(&intAvl, IntTraverse);
56         printf("intAvl traverse in:\n");
57         AvlTravIn(&intAvl, IntTraverse);
58         printf("intAvl traverse post:\n");
59         AvlTravPost(&intAvl, IntTraverse);
60         printf("intAvl traverse level:\n");
61         AvlTravLevel(&intAvl, IntTraverse);
62     }
63     int intRemove = 1;
64     if (0 == AvlRemove(&intAvl, &intRemove))
65     {
66         printf("intAvl remove key %d success\n", intRemove);
67     }
68     else
69     {
70         printf("intAvl remove key %d fail\n", intRemove);
71     }
72     if (0 == AvlRemove(&intAvl, &intRemove))
73     {

```

```

74     printf("intAvl remove key %d success\n", intRemove);
75 }
76 else
77 {
78     printf("intAvl remove key %d fail\n", intRemove);
79 }
80 if (!AvlEmpty(&intAvl))
81 {
82     printf("intAvl size is %d\n", AvlSize(&intAvl));
83     printf("intAvl height is %d\n", AvlHeight(&intAvl));
84     AvlTravInRec(&intAvl, IntTraverse);
85 }
86
87 int intSearch = 2;
88 AVLNODE *node = AvlSearch(&intAvl, &intSearch);
89 if (NULL != node)
90 {
91     printf("key %d is in intAvl\n", intSearch);
92 }
93 else
94 {
95     printf("key %d is not in intAvl\n", intSearch);
96 }
97 intSearch = 11;
98 node = AvlSearch(&intAvl, &intSearch);
99 if (NULL != node)
100 {
101     printf("key %d is in intAvl\n", intSearch);
102 }
103 else
104 {
105     printf("key %d is not in intAvl\n", intSearch);
106 }
107 AvlDispose(&intAvl);
108
109 printf("\n\n");
110
111 AVLTREE stringAvl;
112 AvlNew(&stringAvl, sizeof(char *), StringCmp, StringFree);
113 char *name1 = strdup("pc");
114 char *name2 = strdup("pcwl513");
115 char *name3 = strdup("pcpc");
116 char *name4 = strdup("jerry");
117 char *name5 = strdup("jerry.peng");
118 char *name6 = strdup("yanglupu");
119 char *name7 = strdup("zhanglei");
120 char *name8 = strdup("lishanke");
121 AvlInsert(&stringAvl, &name1);
122 AvlInsert(&stringAvl, &name2);
123 AvlInsert(&stringAvl, &name3);
124 AvlInsert(&stringAvl, &name4);
125 AvlInsert(&stringAvl, &name5);
126 AvlInsert(&stringAvl, &name6);
127 AvlInsert(&stringAvl, &name7);
128 AvlInsert(&stringAvl, &name8);
129 if (!AvlEmpty(&stringAvl))
130 {
131     printf("stringAvl size is %d\n", AvlSize(&stringAvl));
132     printf("stringAvl height is %d\n", AvlHeight(&stringAvl));
133     printf("stringAvl traverse pre:\n");
134     AvlTravPre(&stringAvl, StringTraverse);
135     printf("stringAvl traverse in:\n");
136     AvlTravIn(&stringAvl, StringTraverse);
137     printf("stringAvl traverse post:\n");
138     AvlTravPost(&stringAvl, StringTraverse);
139     printf("stringAvl traverse level:\n");
140     AvlTravLevel(&stringAvl, StringTraverse);
141 }
142
143 char *strRemove = "pcpc";
144 if (0 == AvlRemove(&stringAvl, &strRemove))
145 {
146     printf("stringAvl remove key %s success\n", strRemove);

```

```

147     }
148     else
149     {
150         printf("stringAvl remove key %s fail\n", strRemove);
151     }
152     if (0 == AvlRemove(&stringAvl, &strRemove))
153     {
154         printf("stringAvl remove key %s success\n", strRemove);
155     }
156     else
157     {
158         printf("stringAvl remove key %s fail\n", strRemove);
159     }
160     if (!AvlEmpty(&stringAvl))
161     {
162         printf("stringAvl size is %d\n", AvlSize(&stringAvl));
163         printf("stringAvl height is %d\n", AvlHeight(&stringAvl));
164         AvlTravInRec(&stringAvl, StringTraverse);
165     }
166
167     char *strSearch = "yanglupu";
168     node = AvlSearch(&stringAvl, &strSearch);
169     if (NULL != node)
170     {
171         printf("key %s is in stringAvl\n", strSearch);
172     }
173     else
174     {
175         printf("key %s is not in stringAvl\n", strSearch);
176     }
177     strSearch = "123";
178     node = AvlSearch(&stringAvl, &strSearch);
179     if (NULL != node)
180     {
181         printf("key %s is in stringAvl\n", strSearch);
182     }
183     else
184     {
185         printf("key %s is not in stringAvl\n", strSearch);
186     }
187     AvlDispose(&stringAvl);
188     return 0;
189 }

```