```c
/* OrderVectorTest.c */
#include <stdio.h>
#include <malloc.h>
#include <string.h>
#include <stdlib.h>
#include "OrderVector.h"

static int IntCmp(const void *keyAddr, const void *elemAddr)
{
    int *p1 = (int *)keyAddr;
    int *p2 = (int *)elemAddr;
    return (*p1 - *p2);
}

static void IntTraverse(void *elemAddr, void *outData)
{
    int *p = (int *)elemAddr;
    printf("%d\n", *p);
}

static int StringCmp(const void *keyAddr, const void *elemAddr)
{
    char *p1 = *(char **)keyAddr;
    char *p2 = *(char **)elemAddr;
    return strcmp(p1, p2);
}

static void StringTraverse(void *elemAddr, void *outData)
{
    char *p = *(char **)elemAddr;
    printf("%s\n", p);
}

static void StringFree(void *elemAddr)
{
    free(*(char **)elemAddr);
}

int main()
{
    VECTOR intVector;
    VectorNew(&intVector, sizeof(int), 4, 1, IntCmp, NULL);
    int i = 0;
    for (; i < 10; i ++)
    {
        VectorInsert(&intVector, &i);
    }
    if (!VectorEmpty(&intVector))
    {
        printf("intVector size = %d\n", VectorSize(&intVector));
        VectorTraverse(&intVector, IntTraverse, NULL);
    }
    int intRemove = 5;
    if (0 == VectorRemove(&intVector, &intRemove))
    {
        printf("intVector remove %d success\n", intRemove);
    }
    else
    {
        printf("intVector remove %d fail\n", intRemove);
    }
    if (0 == VectorRemove(&intVector, &intRemove))
    {
        printf("intVector remove %d success\n", intRemove);
    }
    else
    {
        printf("intVector remove %d fail\n", intRemove);
    }
    if (!VectorEmpty(&intVector))
    {
        printf("intVector size = %d\n", VectorSize(&intVector));
        VectorTraverse(&intVector, IntTraverse, NULL);
```

```
    }
    int intSearch = 3;
    int pos = VectorSearch(&intVector, &intSearch, 0);
    if (VectorFind(&intVector, pos, &intSearch))
    {
        printf("the position of data %d in intVector is %d\n", intSearch, pos);
    }
    else
    {
        printf("data %d is not in intVector\n", intSearch);
    }
    intSearch = 11;
    pos = VectorSearch(&intVector, &intSearch, 1);
    if (VectorFind(&intVector, pos, &intSearch))
    {
        printf("the position of data %d in intVector is %d\n", intSearch, pos);
    }
    else
    {
        printf("data %d is not in intVector\n", intSearch);
    }
    VectorMakeEmpty(&intVector);
    if (VectorEmpty(&intVector))
    {
        printf("intVector is made empty success\n");
    }
    else
    {
        printf("intVector is made empty fail\n");
    }

    for (i = 20; i > 10; i --)
    {
        VectorInsert(&intVector, &i);
    }
    if (!VectorEmpty(&intVector))
    {
        printf("intVector size = %d\n", VectorSize(&intVector));
        VectorTraverse(&intVector, IntTraverse, NULL);
    }
    VectorDispose(&intVector);

    printf("\n\n");

    VECTOR stringVector;
    VectorNew(&stringVector, sizeof(char *), 8, 0, StringCmp, StringFree);
    char *name1 = strdup("jerry");
    char *name2 = strdup("pc");
    char *name3 = strdup("pcwl513");
    char *name4 = strdup("pcpc");
    char *name5 = strdup("zhanglei");
    char *name6 = strdup("lishanke");
    char *name7 = strdup("yanglupu");
    char *name8 = strdup("jerry.peng");
    VectorInsert(&stringVector, &name1);
    VectorInsert(&stringVector, &name2);
    VectorInsert(&stringVector, &name3);
    VectorInsert(&stringVector, &name4);
    VectorInsert(&stringVector, &name5);
    VectorInsert(&stringVector, &name6);
    VectorInsert(&stringVector, &name7);
    VectorInsert(&stringVector, &name8);
    if (!VectorEmpty(&stringVector))
    {
        printf("stringVector size = %d\n", VectorSize(&stringVector));
        VectorTraverse(&stringVector, StringTraverse, NULL);
    }
    char *strRemove = "jerry.peng";
    if (0 == VectorRemove(&stringVector, &strRemove))
    {
        printf("stringVector remove %s success\n", strRemove);
    }
    else
```

```
147         {
148             printf("stringVector remove %s fail\n", strRemove);
149         }
150         if (0 == VectorRemove(&stringVector, &strRemove))
151         {
152             printf("stringVector remove %s success\n", strRemove);
153         }
154         else
155         {
156             printf("stringVector remove %s fail\n", strRemove);
157         }
158         if (!VectorEmpty(&stringVector))
159         {
160             printf("stringVector size = %d\n", VectorSize(&stringVector));
161             VectorTraverse(&stringVector, StringTraverse, NULL);
162         }
163         char *strSearch = "zhanglei";
164         pos = VectorSearch(&stringVector, &strSearch, 0);
165         if (VectorFind(&stringVector, pos, &strSearch))
166         {
167                 printf("the position of data %s in stringVector is %d\n", strSearch, pos);
168         }
169         else
170         {
171             printf("data %s is not in stringVector\n", strSearch);
172         }
173         strSearch = "123";
174         pos = VectorSearch(&stringVector, &strSearch, 1);
175         if (VectorFind(&stringVector, pos, &strSearch))
176         {
177                 printf("the position of data %s in stringVector is %d\n", strSearch, pos);
178         }
179         else
180         {
181             printf("data %s is not in stringVector\n", strSearch);
182         }
183         VectorDispose(&stringVector);
184         return 0;
185     }
```