

上机随堂作业

– 教材

P.185 题 1 (编程题, 结构入门)

要求: 1). 通讯录条目的结构体成员至少包含 *name*、*phone_number*、*address*; 2) 分别编写输入和输出一条通讯录条目的函数。

测试样例: 自定即可, 完整测试两个函数功能。

– 补充题: 编写一个程序, 实现一个单词备忘录。

要求: 1) 创建一个内容为空的 struct 体

```
typedef struct{
    char word[20];    /*单词名*/
    int count;        /*单词计数器*/
}KeyTab; KeyTab tab[M];
```

2) 若在查询中找到, 则单词计数器增 1; 若未找到, 字典新增一个新单词, 单词计数器记为 1。

测试样例: 自定即可, 包含找到/未找到两种情况的截图, 并显示其对应的单词计数器数值。

–上机指导书

实验七 思考题 1, 2

吴老师, 骆学长和孙学长好, 以下是我的作业。

T1

P.185 题 1 (编程题, 结构入门)

要求: 1). 通讯录条目的结构体成员至少包含 *name*、*phone_number*、*address*; 2) 分别编写输入和输出一条通讯录条目的函数。

测试样例: 自定即可, 完整测试两个函数功能。

一个简单的 struct 类

```
C Pokers.c M C StudentInfoManagement.c M C T1 addressbook.c X
code_pack > C T1 addressbook.c > add(Contact [], const Contact, int *)
1 //name, phone_number, address;
2 #include<stdio.h>
3 struct Contact
4 {
5     char name[20];
6     char phone_number[15];
7     char address[50];
8 };
9 typedef struct Contact Contact;
10 void add(Contact contacts[],const Contact c,int *pn)
11 {
12     ++(*pn);
13     contacts[*pn]=c;
14     return;
15 }
16 void query(const Contact contacts[],int id,const int *pn)
17 {
18     if(id<1 || id>(*pn))
19     {
20         printf("Invalid ID\n");
21         return;
22     }
23     printf("Name: %s\nPhone Number: %s\nAddress: %s\n", contacts[id].name,
24           contacts[id].phone_number, contacts[id].address);
25 }
26 int main()
27 {
28     int op;
```

```
24 }
25 int main()
26 {
27     int op;
28     int n;
29     Contact contacts[100];
30     while(1)
31     {
32         printf("Input an op:\n");
33         scanf("%d",&op);
34         switch(op)
35         {
36             case 1:
37             {
38                 /*
39                  * 大括号 {} 只是多了一层“块作用域”，里面可以单独定义变量，不会和别的 case 冲突，比如：
40                  */
41                 Contact c;
42                 printf("Input name, phone number, address:\n");
43                 scanf("%s %s %s", c.name, c.phone_number, c.address);
44                 // printf("Name: %s\nPhone Number: %s\nAddress: %s\n", c.name, c.
45                   phone_number, c.address);
46                 add(contacts,c,&n);
47                 break;
48             }
49             case 2:
50             {
51                 int id;
52                 scanf("%d", &id);
53                 query(contacts, id, &n);
54                 break;
55             }
56         }
57     }
58 }
```

• 很普
○
○
○
○
你在这
• 宏:
• 结构
• 函数
que
• 普通
pla
总之:
提供指
Agent

C/C++ Runner: Debug Session (CS10004 Programming) 行 12, 列 13 空格: 4 UTF-8 LF { } C

```
44         printf("Name: %s\nPhone Number: %s\nAddress: %s\n", c.name, c.  
        phone_number, c.address);  
45         add(&contacts, c, &n);  
46         break;  
47     }  
48     case 2:  
49     {  
50         int id;  
51         scanf("%d", &id);  
52         query(&contacts, id, &n);  
53         break;  
54     }  
55     default:  
56     {  
57         printf("Invalid operation\n");  
58         break;  
59     }  
60 }  
61 }  
62 return 0;  
63 }
```

问题 输出 调试控制台 终端 端口 3

Phone Number: 54321
Address: ShanghaiUniversity
Input an op:

```
Input an op:  
1  
Input name, phone number, address:  
Jason 12345 FudanUniversity  
Input an op:  
1  
Input name, phone number, address:  
Lucy 54321 ShanghaiUniversity  
Input an op:  
2  
1  
Name: Jason  
Phone Number: 12345  
Address: FudanUniversity  
Input an op:  
2  
2  
Name: Lucy  
Phone Number: 54321  
Address: ShanghaiUniversity  
Input an op:  
1
```

t2

2) 分别编写输入和输出一条通讯录条目的函数。

测试样例：自定即可，完整测试两个函数功能。

- 补充题：编写一个程序，实现一个单词备忘录。

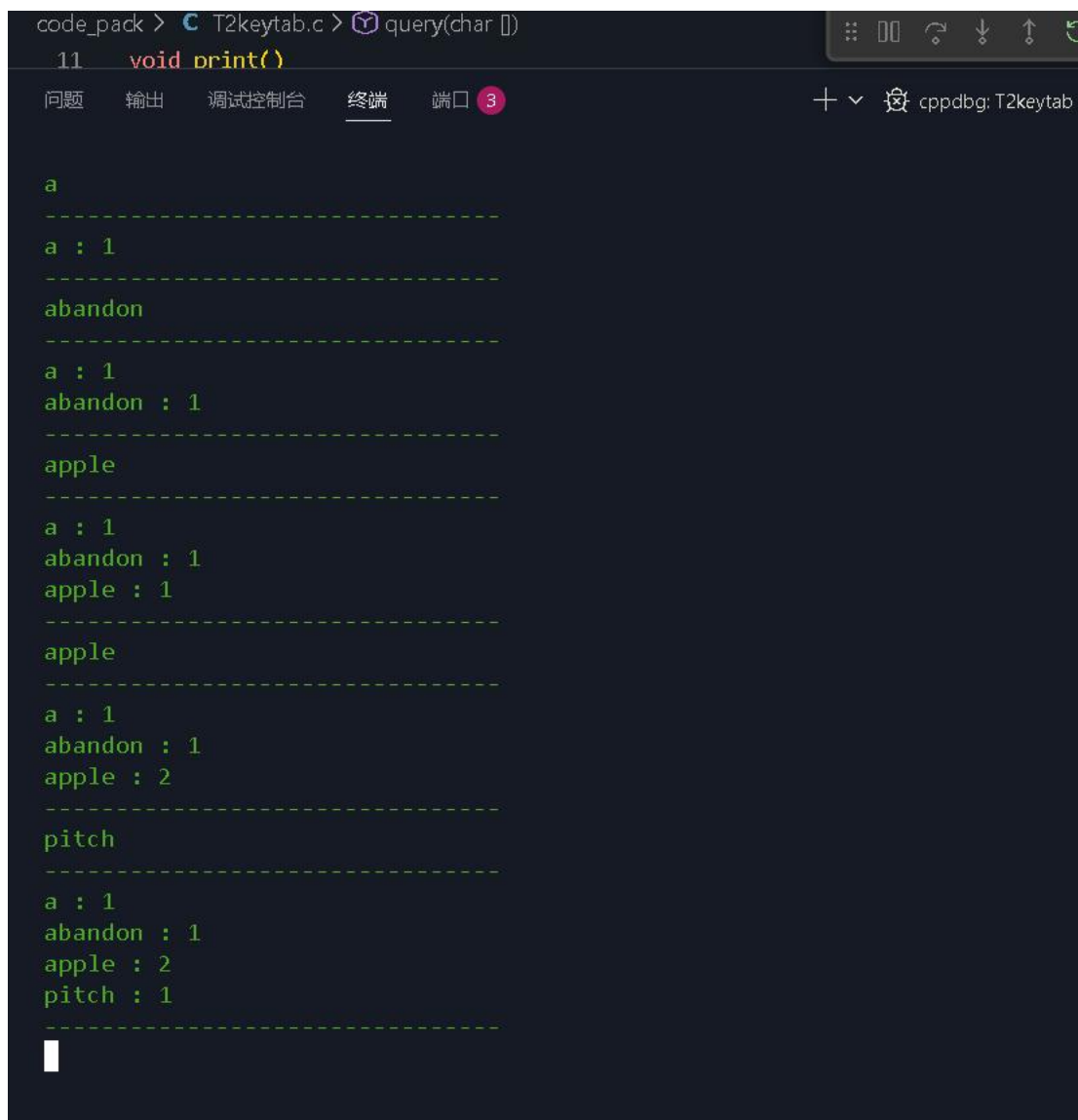
要求：1) 创建一个内容为空的结构体

```
typedef struct{  
    char word[20];    /*单词名*/  
    int count;        /*单词计数器*/  
}KeyTab; KeyTab tab[M];
```

2) 若在查询中找到, 则单词计数器增 1; 若未找到, 字典新增一个新单词, 单词计数器记为 1。

测试样例: 自定即可, 包含找到/未找到两种情况的截图, 并显示其对应的单词计数器数值。

通过遍历来找单词 ()



```
code_pack > T2keytab.c > query(char [])
11 void print()

问题  输出  调试控制台  终端  端口 3  + v  cppdbg: T2keytab

a
-----
a : 1
-----
abandon
-----
a : 1
abandon : 1
-----
apple
-----
a : 1
abandon : 1
apple : 1
-----
apple
-----
a : 1
abandon : 1
apple : 2
-----
pitch
-----
a : 1
abandon : 1
apple : 2
pitch : 1
-----
█
```

```

pack > T2keytab.c > main()
#include<stdio.h>
#include<string.h>
# define M 100
typedef struct
{
    char word[2000];    /*单词名*/
    int count;          /*单词计数器*/
}KeyTab;
KeyTab tab[M];
int n;
void print()
{
    printf("-----\n");
    for(int k=1;k<=n;++k)
    {
        printf("%s : %d\n",tab[k].word,tab[k].count);
    }
    printf("-----\n");
}
void query(char myword[])
{
    for(int i=1;i<=n;++i)
    {
        if(strcmp(myword,tab[i].word)==0)
        {
            tab[i].count++;
            return;
        }
    }
}

```

```

ode_pack > T2keytab.c > main()
11 void print()
14     for(int k=1;k<=n;++k)
15     {
16         printf("%s : %d\n",tab[k].word,tab[k].count);
17     }
18     printf("-----\n");
19 }
20 void query(char myword[])
21 {
22     for(int i=1;i<=n;++i)
23     {
24         if(strcmp(myword,tab[i].word)==0)
25         {
26             tab[i].count++;
27             return;
28         }
29     }
30     ++n;
31     tab[n].count=1;
32     strcpy(tab[n].word,myword);//这里不能直接赋值，这里是两个数组
33 }
34 int main()
35 {
36     char myword[100];
37     while(1)
38     {
39         scanf("%s",myword);

```

```

24         if(strcmp(myword,tab[i].word)==0)
25         {
26             //这里不能直接赋值，这里是两个数组
27             strcpy(tab[i].word,myword);
28         }
29     }
30     ++n;
31     tab[n].count=1;
32     strcpy(tab[n].word,myword);
33 }
34 int main()
35 {
36     char myword[100];
37     while(1)
38     {
39         scanf("%s",myword);
40         query(myword);
41         print();
42     }
43     return 0;
44 }
45

```

3. 类似于周末作业，维护一个类和一些相关的函数

实验七 思考题 1

```

studentInfoManagement.c M  C T3 exp1.c  C T1 addressbook.c  C T2keytab.c  聊天
code_pack > C T3 exp1.c > main()
1  #include<stdio.h>
2  #include<string.h>
3  #include<stdlib.h>
4
5  #define MAX_NUM_OF_ALL 50
6
7  struct Student
8  {
9      char name[MAX_NUM_OF_ALL];
10     float score[MAX_NUM_OF_ALL];
11     float total;
12 };
13 typedef struct Student Student;
14 void Input_StudentInfo(Student s[],int n,const int CourseNum)
15 {
16     for(int i=0;i<n;i++)
17     {
18         s[i].total=0.0f;
19         scanf("%s",s[i].name);
20         for(int j=0;j<CourseNum;j++)
21         {
22             scanf("%f",&s[i].score[j]);
23             s[i].total+=s[i].score[j];
24         }
25     }
26 }
27
28

```

```

code_pack > T3 exp1.c > main()
4   void Input_StudentInfo(Student s[],int n,const int CourseNum)
6       for(int i=0;i<n;i++)
6
7   }
8
9   void Query_CourseInfo(const Student s[],const int n,const int CourseNum)
10  {
11      int c;
12      c=0;
13
14      printf("Chinese\n");
15      float sum=0.0f;
16      for(int i=0;i<n;i++)sum+=s[i].score[c];
17      printf("sum=%.2f avg=%.2f\n",sum,sum/n);
18
19      printf("*****\n");
20
21      c=1;
22      printf("Math\n");
23      sum=0.0f;
24      for(int i=0;i<n;i++)sum+=s[i].score[c];
25      printf("sum=%.2f avg=%.2f\n",sum,sum/n);
26  }
27
28
29  int main()
30  {
31      Student s[MAX_NUM_OF_ALL];
32      int n;
33      int CourseNum;
34      n=3;

```

```

code_pack > T3 exp1.c > main()
29   void Query_CourseInfo(const Student s[],const int n,const int CourseNum)
44       for(int i=0;i<n;i++)sum+=s[i].score[c];
45       printf("sum=%.2f avg=%.2f\n",sum,sum/n);
46   }
47
48
49   int main()
50   {
51       Student s[MAX_NUM_OF_ALL];
52       int n;
53       int CourseNum;
54       n=3;
55       CourseNum=2;
56       Input_StudentInfo(s,n,CourseNum);
57
58       Query_CourseInfo(s, n,CourseNum);
59       return 0;
60   }
61

```

问题 输出 调试控制台 终端 端口 3

```

aaa 100 100
bbb 88 90
ccc 77 7
Chinese
sum=265.00 avg=88.33
*****
Math
sum=197.00 avg=65.67

```

T4

实验七 思考题 1, 2

输出

(op 1 是加名字 op 2 是打印 op3 是输入编号删去那个点)
多创建了一个虚拟头和虚拟尾, 让删除函数形式上比较好看

```
#####
```

```
op:?
```

```
1
```

```
Input id, name, score:
```

```
1 jason 100
```

```
Successfully Append;
```

```
#####
```

```
op:?
```

```
1
```

```
Input id, name, score:
```

```
5 lucy 99
```

```
Successfully Append;
```

```
#####
```

```
op:?
```

```
1
```

```
Input id, name, score:
```

```
9 tom 99
```

```
Successfully Append;
```

```
#####
```

```
op:?
```

```
2
```

1 jason 100.00

5 lucy 99.00

9 tom 99.00

#####

op:?

3

Input id

9

Successfully Delete;

#####

op:?

2

1 jason 100.00

5 lucy 99.00

#####

op:?

3

Input id

1

Successfully Delete;

#####

op:?

2

5 lucy 99.00

#####

op:?

3

Input id

5

Successfully Delete;

#####

op:?

2

#####

op:?

1

Input id, name, score:

10 Gilbert 1000

Successfully Append;

#####

op:?

2

10 Gilbert 1000.00

#####

op:?

```
op:?
1

Input id, name, score:
5 lucy 99
Successfully Append;
#####
op:?
1

Input id, name, score:
9 tom 99
Successfully Append;
#####
op:?
2

*****
1 jason 100.00
5 lucy 99.00
9 tom 99.00
*****
#####
op:?
3

Input id
9
Successfully Delete;
#####
op:?
2

*****
1 jason 100.00
5 lucy 99.00
```

```

ack > C LinkedLists T4.c > append(Student *, Student *, const Student)
#include<stdio.h>
#include<stdlib.h>
struct Student
{
    int id;
    char name[100];
    float score;
    struct Student *nxt;
};
typedef struct Student Student;
void append(Student* pStart, Student* pEnd, const Student newstudent)
{
    Student* ptr;
    ptr=pStart;
    while(ptr->nxt!=pEnd)ptr=ptr->nxt;
    (ptr->nxt)=(Student*)malloc(sizeof(Student));
    *(ptr->nxt)=newstudent;
    (ptr->nxt->nxt)=pEnd;
    printf("Successfully Append;\n");
}
void print(Student* pStart, Student* pEnd)
{
    printf("*****\n");
    Student* ptr;
    ptr=pStart;
    while(ptr->nxt!=pEnd)
    {
        ptr=ptr->nxt; // 先跳了再说
        printf("%d %s %.2f\n", ptr->id, ptr->name, ptr->score);
    }
}

```

值得注意的是，这里传 start 必须传指针，因为它相当于 head，会被更改

```

ack > C LinkedLists T4.c > append(Student *, Student *, const Student)
void print(Student* pStart, Student* pEnd)
{
    printf("*****\n");
    Student* ptr;
    ptr=pStart;
    while(ptr->nxt!=pEnd)
    {
        ptr=ptr->nxt; // 先跳了再说
        printf("%d %s %.2f\n", ptr->id, ptr->name, ptr->score);
    }
    printf("*****\n");
}
void del(Student* pStart, Student* pEnd, int oldid)
{
    Student* ptr;
    ptr=pStart;
    while(ptr->nxt!=pEnd)
    {
        if((ptr->nxt)->id==oldid)
        {
            ptr->nxt=(ptr->nxt)->nxt;
            printf("Successfully Delete;\n");
            return;
        }
        ptr=ptr->nxt;
    }
}
int main()
{
    Student Start, End; // 两个虚拟的空头空尾
    Start.nxt=&End;
    End.nxt=NULL;
}

```

```

int main()
{
    Student Start,End;//两个虚拟的空头空尾
    Start.next=&End;
    End.next=NULL;

    int n;
    int op;
    while(1)
    { printf("#####\n op:? \n");
      scanf("%d",&op);
      printf("\n");
      switch(op)
      {
          case 1:
          {
              printf("Input id, name, score:\n");
              Student newstudent;
              scanf("%d %s %f",&newstudent.id,newstudent.name,&newstudent.score);
              append(&Start,&End,newstudent);
              break;
          }
          case 2:
          {
              print(&Start,&End);
              break;
          }
          case 3:
          {
              printf("Input id\n");

```

```

de_pack > C: LinkedLists T4.c > append(Student *, Student *, const Student)
51     switch(op)
52     {
53     {
54         case 1:
55         {
56             printf("Input id, name, score:\n");
57             Student newstudent;
58             scanf("%d %s %f",&newstudent.id,newstudent.name,&newstudent.score);
59             append(&Start,&End,newstudent);
60             break;
61         }
62         case 2:
63         {
64             print(&Start,&End);
65             break;
66         }
67         case 3:
68         {
69             printf("Input id\n");
70             int oldid;
71             scanf("%d",&oldid);
72             del(&Start,&End,oldid);
73             break;
74         }
75         default:
76             return 0;
77     }
78 }
79 return 0;
80 }
81
82
83
84
85
86
87
88
89
90

```