

温州大学计算机与人工智能学院

程序设计课程设计 实验报告

实验名称	动态链表：查找				
班 级	23大数据1	姓 名	徐王晶	学 号	23211870102
实验地点	南5B105	实验时间	2023-12-26,14:19:06	指导老师	虞铭财

一、问题编号：

2079

地址：<http://10.132.254.54/problem/2079/>

二、问题描述：

请设计一个简单的学生成绩管理系统，要求系统实现以下功能：
学生信息包括学号、姓名、性别、语文、数学、英语。
插入学生信息：
Insert id name sex x y z 其中的参数分别为学号、姓名、性别、三门课的成绩，成绩为浮点数。
输出所有学生信息：
List
按照插入的顺序输出所有学生的信息，每行一位学生的信息。每行的格式如下：
id name sex x y z
数据之间有一个空格分隔，成绩保留1位小数。
查找学生信息：
Find id
查找学号为id的学生信息。
退出程序：
Quit或者Exit

三、输入说明：

输入有多行，每行一条指令，指令格式如下
Insert id name sex x y z
插入学生信息，分别为学号、姓名、性别和三门课的成绩。
List
输出所有学生信息。
Find id
查找学号为id的学生信息。
Quit或者Exit
输出"Good bye!"后结束程序。

四、输出说明：

输出有多行，对应命令的输出如下：
Insert id name sex x y z
插入后在先在单独的一行中输出"Insert:"，然后在第二行中显示学生信息，数据之间用一个空格分开，成绩保留1位小数。
List
第一行输出"List:"，接下来按照插入的顺序输出所有学生的信息，每行一位学生的信息。每行的格式如下：
id name sex x y z
数据之间用一个空格分开，成绩保留1位小数。
Find id
第一行显示"Find:"，第二行显示格式如下：
如果找到学号为id的学生，则在单独一行中显示学生信息，格式如List。否则在单独一行显示"Failed"。
Quit或者Exit
在单独一行中输出"Good bye!"后结束程序。

五、输入样例：

```
Insert 0911001 zhangsan F 87 78 65
Insert 0911002 zhaoliu F 97 90 55
Insert 0911003 Lisi F 77 72 55
Find 0911004
Insert 0911004 Wangwu F 68 56 95
Find 0911004
List
Quit
```

六、输出样例：

```
Insert:
0911001 zhangsan F 87.0 78.0 65.0
Insert:
0911002 zhaoliu F 97.0 90.0 55.0
Insert:
0911003 Lisi F 77.0 72.0 55.0
Find:
Failed
Insert:
0911004 Wangwu F 68.0 56.0 95.0
Find:
0911004 Wangwu F 68.0 56.0 95.0
List:
0911001 zhangsan F 87.0 78.0 65.0
0911002 zhaoliu F 97.0 90.0 55.0
0911003 Lisi F 77.0 72.0 55.0
0911004 Wangwu F 68.0 56.0 95.0
Good bye!
```

七、解答内容：

所用语言：

源代码：

```
001. #include <stdio.h>
002. #include <string.h>
003. #include <stdlib.h>
004.
005. struct Data
006. {
007.     char id[20];
008.     char name[20];
009.     char sex;
010.     double x;
011.     double y;
012.     double z;
013.     struct Data *next;
014. };
015.
016. struct LinkList
017. {
018.     struct Data *head;
019. };
020.
021. void Insert(struct LinkList *llst);
022. void List(struct LinkList *llst);
023. void Find(struct LinkList *llst);
024.
025. int main(void)
026. {
027.     //freopen("./in1.txt", "r", stdin);
028.
029.     char op[20];
030.     struct LinkList llst;
031.     llst.head = NULL;
032.
033.     while (scanf("%s", op), strcmp(op, "Quit") != 0)
034.     {
```

```

035.     if (strcmp(op, "Insert") == 0)
036.     {
037.         printf("Insert:\n");
038.         Insert(&llst);
039.     }
040.     else if (strcmp(op, "List") == 0)
041.     {
042.         printf("List:\n");
043.         List(&llst);
044.     }
045.     else if (strcmp(op, "Find") == 0)
046.     {
047.         printf("Find:\n");
048.         Find(&llst);
049.     }
050.     else if (strcmp(op, "Change") == 0)
051.     {
052.         printf("Change:\n");
053.     }
054.     else if (strcmp(op, "Delete") == 0)
055.     {
056.         printf("Delete:\n");
057.     }
058. }
059. printf("Good bye!\n");
060.
061. return 0;
062. }
063.
064. void Insert(struct LinkList *llst)
065. {
066.     struct Data * node = (struct Data *)malloc(sizeof(struct Data));
067.     scanf(
068.         "%s %s %c %lf %lf %lf",
069.         node -> id,
070.         node->name,
071.         &(node -> sex),
072.         &(node -> x),
073.         &(node -> y),
074.         &(node -> z)
075.     );
076.     node -> next = NULL;
077.
078.     struct Data * p = llst -> head;
079.     struct Data * q = p;
080.     if (p == NULL)
081.     {
082.         llst -> head = node;
083.     }
084.     else
085.     {
086.         while (p != NULL)
087.         {
088.             q = p;
089.             p = p -> next;
090.         }
091.         q -> next = node;
092.     }
093.
094.     printf(
095.         "%s %s %c %.1lf %.1lf %.1lf\n",
096.         node -> id,
097.         node -> name,
098.         node -> sex,
099.         node -> x,
100.         node -> y,
101.         node -> z
102.     );
103. }
104.
105. void List(struct LinkList *llst)
106. {
107.     struct Data * p = llst -> head;
108.     while (p != NULL)
109.     {
110.         printf(
111.             "%s %s %c %.1lf %.1lf %.1lf\n",
112.             p -> id,
113.             p -> name,
114.             p -> sex,
115.             p -> x,
116.             p -> y,
117.             p -> z
118.         );
119.         p = p -> next;
120.     }
121. }
122.
123. void Find(struct LinkList *llst)
124. {
125.     char id[20];

```

```
126. scanf("%s", id);
127.
128. struct Data * p = llst -> head;
129. while (p != NULL && strcmp(p -> id, id) != 0)
130. {
131.     p = p -> next;
132. }
133.
134. if (p != NULL)
135. {
136.     printf(
137.         "%s %s %c %.11f %.11f %.11f\n",
138.         p -> id,
139.         p -> name,
140.         p -> sex,
141.         p -> x,
142.         p -> y,
143.         p -> z
144.     );
145. }
146. else
147. {
148.     printf("Failed\n");
149. }
150. }
```

八、判题结果

AC - 答案正确

判题结果补充说明:

test id:4019,result:AC, usedtime:0MS, usedmem:864KB,score:50 test id:4020,result:AC, usedtime:0MS,
usedmem:864KB,score:50