

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：排序				
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	2023-12-26,14:44:35	指导老师	虞铭财

一、问题编号：

0538

地址: <http://10.132.254.54/problem/538/>

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。如果链表中已经存在相同学号的学生信息，插入失败。
输出所有学生信息：
List
按照学号从小到大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。
修改学生信息：
Change id newname, newx, newy, newz
把学号为id的学生信息修改为newname, newx, newy, newz（学号保持不变）
删除学生信息：
Delete id
删除学号为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
按照学号从小到大的输出所有学生信息。
Find id
查找学号为id的学生信息。
Change id newname, newx, newy, newz
把学号为id的学生信息修改为newname, newx, newy, newz（学号保持不变）
Delete id
删除学号为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
    插入后先在单独的一行输出"Insert:"。
    第二行中显示信息如下：如果链表中已经存在相同学号的学生信息，插入失败，显示"Failed"。否则插入并显示学生信息，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:"，接下来按照学号从小到大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
    数据之间一个空格，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"。
Change id newname, newx, newy, newz
    第一行显示"Change:"。如果链表中不存在学号为id的学生，显示"Failed"。否则修改该学生信息并在单独一行中显示该生信息，显示格式如List命令。
Delete id
    第一行显示"Delete:"。如果链表中不存在学号为id的学生，显示"Failed"。否则修改该学生信息并在单独一行中显示"Deleted"
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 0911002 zhaoliu  F 97 90 55
Insert 0911001 zhangsan  F 87 78 65
Insert 0911001 zhangsan  F 87 78 65
Insert 0911003 Lisi    F 77 72 55
Change 0911001 Zhangsan  M 77 78 65
Change 0911004 Wangwu   M 77 78 65
Insert 0911004 Wangwu   F 68 56 95
Find 0911004
List
Delete 0911004
Delete 0911004
List
Quit
```

六、输出样例：

```
Insert:
0911002 zhaoliu F 97.0 90.0 55.0
Insert:
0911001 zhangsan F 87.0 78.0 65.0
Insert:
Failed
Insert:
0911003 Lisi F 77.0 72.0 55.0
Change:
0911001 Zhangsan M 77.0 78.0 65.0
Change:
Failed
Insert:
0911004 Wangwu F 68.0 56.0 95.0
Find:
0911004 Wangwu F 68.0 56.0 95.0
List:
0911001 Zhangsan M 77.0 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
Delete:
Deleted
Delete:
Failed
List:
0911001 Zhangsan M 77.0 78.0 65.0
0911002 zhaoliu F 97.0 90.0 55.0
0911003 Lisi F 77.0 72.0 55.0
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. void Change(struct LinkList *llst);
025. void Delete(struct LinkList *llst);
026.
027. int main(void)
028. {
029.     //freopen("./in1.txt", "r", stdin);
030.
031.     char op[20];
032.     struct LinkList llst;
033.     llst.head = NULL;
034.
035.     while (scanf("%s", op), strcmp(op, "Quit") != 0)
036.     {
037.         if (strcmp(op, "Insert") == 0)
038.         {
039.             printf("Insert:\n");
040.             Insert(&llst);
041.         }
042.         else if (strcmp(op, "List") == 0)
043.         {
```

```

044.         printf("List:\n");
045.         List(&llst);
046.     }
047.     else if (strcmp(op, "Find") == 0)
048.     {
049.         printf("Find:\n");
050.         Find(&llst);
051.     }
052.     else if (strcmp(op, "Change") == 0)
053.     {
054.         printf("Change:\n");
055.         Change(&llst);
056.     }
057.     else if (strcmp(op, "Delete") == 0)
058.     {
059.         printf("Delete:\n");
060.         Delete(&llst);
061.     }
062. }
063. printf("Good bye!\n");
064.
065. return 0;
066. }
067.
068. void Insert(struct LinkList *llst)
069. {
070.     struct Data * node = (struct Data *)malloc(sizeof(struct Data));
071.     scanf(
072.         "%s %s %c %lf %lf %lf",
073.         node -> id,
074.         node->name,
075.         &(node -> sex),
076.         &(node -> x),
077.         &(node -> y),
078.         &(node -> z)
079.     );
080.     node -> next = NULL;
081.
082.     struct Data * p = llst -> head;
083.     struct Data * q = p;
084.     while (p != NULL && strcmp(p -> id, node -> id) < 0)
085.     {
086.         q = p;
087.         p = p -> next;
088.     }
089.
090.     if (p != NULL && strcmp(p -> id, node -> id) == 0)
091.     {
092.         printf("Failed\n");
093.         return;
094.     }
095.
096.     if (p == llst -> head)
097.     {
098.         node -> next = llst -> head;
099.         llst -> head = node;
100.     }
101.     else
102.     {
103.         q -> next = node;
104.         node -> next = p;
105.     }
106.
107.     printf(
108.         "%s %s %c %.11f %.11f %.11f\n",
109.         node -> id,
110.         node -> name,
111.         node -> sex,
112.         node -> x,
113.         node -> y,
114.         node -> z
115.     );
116. }
117.
118. void List(struct LinkList *llst)
119. {
120.     struct Data * p = llst -> head;
121.     while (p != NULL)
122.     {
123.         printf(
124.             "%s %s %c %.11f %.11f %.11f\n",
125.             p -> id,
126.             p -> name,
127.             p -> sex,
128.             p -> x,
129.             p -> y,
130.             p -> z
131.         );
132.         p = p -> next;
133.     }
134. }

```

```

135.
136. void Find(struct LinkList *llst)
137. {
138.     char id[20];
139.     scanf("%s", id);
140.
141.     struct Data * p = llst -> head;
142.     while (p != NULL && strcmp(p -> id, id) != 0)
143.     {
144.         p = p -> next;
145.     }
146.
147.     if (p != NULL)
148.     {
149.         printf(
150.             "%s %s %c %.11f %.11f %.11f\n",
151.             p -> id,
152.             p -> name,
153.             p -> sex,
154.             p -> x,
155.             p -> y,
156.             p -> z
157.         );
158.     }
159.     else
160.     {
161.         printf("Failed\n");
162.     }
163. }
164.
165. void Change(struct LinkList *llst)
166. {
167.     char id[20];
168.     scanf("%s", id);
169.
170.     struct Data * p = llst -> head;
171.     while (p != NULL && strcmp(p -> id, id) != 0)
172.     {
173.         p = p -> next;
174.     }
175.
176.     if (p != NULL)
177.     {
178.         scanf(
179.             "%s %c %lf %lf %lf",
180.             p -> name,
181.             &(p -> sex),
182.             &(p -> x),
183.             &(p -> y),
184.             &(p -> z)
185.         );
186.
187.         printf(
188.             "%s %s %c %.11f %.11f %.11f\n",
189.             p -> id,
190.             p -> name,
191.             p -> sex,
192.             p -> x,
193.             p -> y,
194.             p -> z
195.         );
196.     }
197.     else
198.     {
199.         printf("Failed\n");
200.     }
201. }
202.
203. void Delete(struct LinkList *llst)
204. {
205.     char id[20];
206.     scanf("%s", id);
207.
208.     struct Data * p = llst -> head;
209.     struct Data * q = p;
210.     while (p != NULL && strcmp(p -> id, id) != 0)
211.     {
212.         q = p;
213.         p = p -> next;
214.     }
215.
216.     if (p != NULL)
217.     {
218.         if (p == llst -> head)
219.         {
220.             llst -> head = p -> next;
221.         }
222.         else
223.         {
224.             q -> next = p -> next;
225.         }

```

```
226.         free(p);
227.         printf("Deleted\n");
228.     }
229.     else
230.     {
231.         printf("Failed\n");
232.     }
233. }
```

八、判题结果

AC - 答案正确

判题结果补充说明：

test id:1038,result:AC, usedtime:0MS, usedmem:864KB,score:30 test id:1039,result:AC, usedtime:0MS,
usedmem:864KB,score:30 test id:10210,result:AC, usedtime:0MS, usedmem:864KB,score:40