

# 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：删除				
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	2023-12-26,14:27:36	指导老师	虞铭财

### 一、问题编号：

0537

地址：http://10.132.254.54/problem/537/

### 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 数据之间一个空格，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, newsex, newx, newy, newz  
把学号为id的学生信息修改为newname, newsex, 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 输出所有学生信息。  
Find id 查找学号为id的学生信息。  
Change id newname, newsex, newx, newy, newz 把学号为id的学生信息修改为newname, newsex, 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, newsex, 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"。否则删除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 0911001 zhangsan F 87 78 65
Insert 0911002 zhaoliu F 97 90 55
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
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
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
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.     if (p == NULL)
085.     {
086.         llst -> head = node;
087.     }
088.     else
089.     {
090.         while (p != NULL)
091.         {
092.             q = p;
093.             p = p -> next;
094.         }
095.         q -> next = node;
096.     }
097.
098.     printf(
099.         "%s %s %c %.1lf %.1lf %.1lf\n",
100.         node -> id,
101.         node -> name,
102.         node -> sex,
103.         node -> x,
104.         node -> y,
105.         node -> z
106.     );
107. }
108.
109. void List(struct LinkList *llst)
110. {
111.     struct Data * p = llst -> head;
112.     while (p != NULL)
113.     {
114.         printf(
115.             "%s %s %c %.1lf %.1lf %.1lf\n",
116.             p -> id,
117.             p -> name,
118.             p -> sex,
119.             p -> x,
120.             p -> y,
121.             p -> z
122.         );
123.         p = p -> next;
124.     }
125. }
126.
127. void Find(struct LinkList *llst)
128. {
129.     char id[20];
130.     scanf("%s", id);
131.
132.     struct Data * p = llst -> head;
133.     while (p != NULL && strcmp(p -> id, id) != 0)
134.     {
135.         p = p -> next;
136.     }

```

```

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

```

## 八、判题结果

**AC - 答案正确**

判题结果补充说明：

test id:1036,result:AC, usedtime:0MS, usedmem:864KB,score:50 test id:1037,result:AC, usedtime:0MS,  
usedmem:864KB,score:50