计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名：杨佳森 学号：\_2112080106 专业：数据科学与大数据技术\_年级：\_2021\_

课程：数据结构 主讲教师：\_刘成\_ 辅导教师：\_\_无\_

实验时间：2023年 \_3\_月 \_14\_日 下午17\_时至\_19\_时，实验地点\_计算机大楼213\_

实验题目： 线性表的综合应用 —— 单位员工通讯录管理系统

实验目的： 通过该实验，使学生能灵活掌握对线性表这一数据结构的综合应用，体会在实际应用开发过程中线性表的应用场景和作用。

实验环境（硬件和软件） PC和codeblocks

实验内容：

为某单位建立一个基于文件存储的员工通讯录管理系统，可以方便查询每个员工的办公电话、手机号、电子邮箱等信息。基本功能包括从通讯录文件中读取信息构建通讯录表（顺序表、链表均可），员工信息查询、修改、删除，整个通讯录显示输出，保存等。可以再基本功能基础上，自行扩充。

实验步骤：

1、实现存储结构

typedef struct Staff {

 char Num[5];      *//员工编号*

 char Name[10];    *//员工姓名*

 char Tel[15];     *//办公电话*

 char Phone[15];   *//手机号*

 Staff \*next;

}Staff,\*Sta;

2、链表初始化

void InitList(Sta &s) {

    s = new Staff;

    s->next = NULL;

    cout << "初始化成功！" << endl;

}

3、查询：输入员工编号，输出员工信息，若员工不存在，输出“NOT FOUND!”

void Query(Sta &s) {

 char num[5];

 cout << "请输入员工编号：";

 cin >> num;

 Staff \*sta = new Staff;

 sta = s->next;

 while (sta)

 {

  if (strcmp(sta->Num, num) == 0) {

   cout << "员工姓名：" << sta->Name << endl;

   cout << "办公电话：" << sta->Tel << endl;

   cout << "手机号：" << sta->Phone << endl;

   return;

  }

  sta = sta->next;

 }

 cout << "NOT FOUND!" << endl;

}

4、修改：输入员工编号、要修改的属性、修改信息。姓名为Name、办公电话为Tel、手机号Phone。

void Modify(Sta &s) {

 Staff \*sta = new Staff;

 char num[5],type[20],change[20];

 sta = s->next;

 cin >> num;

 while (sta)

 {

  if (strcmp(sta->Num, num) == 0) {

   cout << "请输入要更改的信息：";

   cin >> type;

   if (strcmp(type, "Name") == 0) {

    cout << "请输入新的员工姓名：";

    cin >> change;

    strcpy(sta->Name, change);

    return;

   }

   else if (strcmp(type, "Tel") == 0) {

    cout << "请输入新的办公电话：";

    cin >> change;

    strcpy(sta->Tel, change);

    return;

   }

   else {

    cout << "请输入新的手机号：";

    cin >> change;

    strcpy(sta->Phone, change);

    return;

   }

  }

  sta = sta->next;

 }

}

5、添加：输入员工编号、姓名、办公电话、手机号。

void Add(Sta &s) {

 Staff \*r = new Staff;

 r = s;

 while (r->next) r = r->next;

 Staff \*sta = new Staff;

 sta->next = NULL;

 cout << "请输入要添加的员工编号，姓名，办公电话，手机号：";

 cin >> sta->Num;

 cin >> sta->Name;

 cin >> sta->Tel;

 cin >> sta->Phone;

 r->next = sta;

}

6、删除：输入员工编号，将其从通讯录中删除。

void Del(Sta &s) {

 char Num[5];

 Staff \*p = new Staff;

 Staff \*q = new Staff;

 p = s;

 cin >> Num;

 while (p)

 {

  if (strcmp(Num, p->next->Num) == 0) {

   q = p->next;

   p->next = q->next;

   delete q;

   return;

  }

  p = p->next;

 }

}

7、打印：输出通讯录。

void PrintStaff(Sta &s) {

 Staff \*p = new Staff;

 p = s->next;

 while (p)

 {

  cout << p->Num << ' ' << p->Name << ' ' << p->Tel << ' ' << p->Phone << endl;

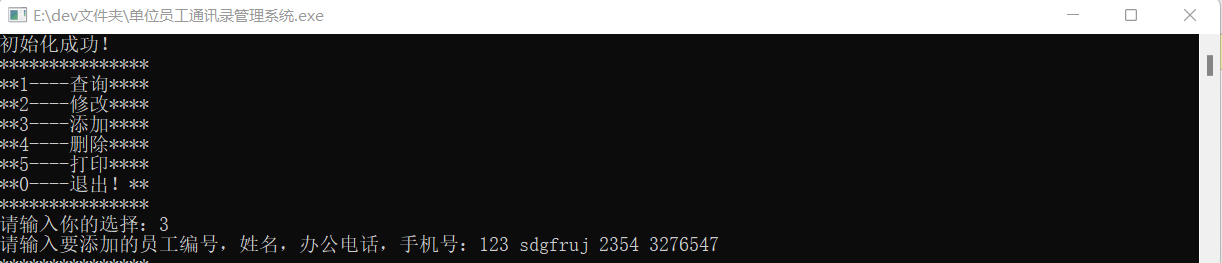
  p = p->next;

 }

}

实验数据记录：

1、初始化通讯录，并添加信息



2、查询通讯录



3、打印通讯录



4、修改手机号，并打印



5、删除并查询



问题讨论：

1. 在输入员工姓名，编号，工作电话和手机号时应注意所开数组大小，不能越界。
2. 通讯录的本质是链表，在使用前应该进行初始化。

全部代码：

1. #include <bits/stdc++.h>
2. using namespace std;
3. typedef struct Staff {
4. char Num[5];      *//员工编号*
5. char Name[10];    *//员工姓名*
6. char Tel[15];     *//办公电话*
7. char Phone[15];   *//手机号*
8. Staff \*next;
9. }Staff,\*Sta;
10. *//初始化*
11. void InitList(Sta &s) {
12. s = new Staff;
13. s->next = NULL;
14. cout << "初始化成功！" << endl;
15. }
16. *//1. 查询：输入员工编号，输出员工信息，若员工不存在，输出“NOT FOUND!”*
17. void Query(Sta &s) {
18. char num[5];
19. cout << "请输入员工编号：";
20. cin >> num;
21. Staff \*sta = new Staff;
22. sta = s->next;
23. while (sta)
24. {
25. if (strcmp(sta->Num, num) == 0) {
26. cout << "员工姓名：" << sta->Name << endl;
27. cout << "办公电话：" << sta->Tel << endl;
28. cout << "手机号：" << sta->Phone << endl;
29. return;
30. }
31. sta = sta->next;
32. }
33. cout << "NOT FOUND!" << endl;
34. }
35. *//2. 修改：输入员工编号、要修改的属性、修改信息。姓名为Name、办公电话为Tel、手机号Phone。*
36. void Modify(Sta &s) {
37. Staff \*sta = new Staff;
38. char num[5],type[20],change[20];
39. sta = s->next;
40. cin >> num;
41. while (sta)
42. {
43. if (strcmp(sta->Num, num) == 0) {
44. cout << "请输入要更改的信息：";
45. cin >> type;
46. if (strcmp(type, "Name") == 0) {
47. cout << "请输入新的员工姓名：";
48. cin >> change;
49. strcpy(sta->Name, change);
50. return;
51. }
52. else if (strcmp(type, "Tel") == 0) {
53. cout << "请输入新的办公电话：";
54. cin >> change;
55. strcpy(sta->Tel, change);
56. return;
57. }
58. else {
59. cout << "请输入新的手机号：";
60. cin >> change;
61. strcpy(sta->Phone, change);
62. return;
63. }
64. }
65. sta = sta->next;
66. }
67. }
68. *// 3. 添加：输入员工编号、姓名、办公电话、手机号。*
69. void Add(Sta &s) {
70. Staff \*r = new Staff;
71. r = s;
72. while (r->next) r = r->next;
73. Staff \*sta = new Staff;
74. sta->next = NULL;
76. cout << "请输入要添加的员工编号，姓名，办公电话，手机号：";
77. cin >> sta->Num;
78. cin >> sta->Name;
79. cin >> sta->Tel;
80. cin >> sta->Phone;
81. *// scanf("%s%s%s%s",&sta->Num,&sta->Name,&sta->Tel,&sta->Phone);*
82. r->next = sta;
83. }
84. *//4. 删除：输入员工编号，将其从通讯录中删除。*
85. void Del(Sta &s) {
86. char Num[5];
88. Staff \*p = new Staff;
89. Staff \*q = new Staff;
91. p = s;
92. cin >> Num;
94. while (p)
95. {
96. if (strcmp(Num, p->next->Num) == 0) {
97. q = p->next;
98. p->next = q->next;
99. delete q;
100. return;
101. }
102. p = p->next;
103. }
104. }
105. *//5. 打印：输出通讯录。*
106. void PrintStaff(Sta &s) {
107. Staff \*p = new Staff;
108. p = s->next;
109. while (p)
110. {
111. cout << p->Num << ' ' << p->Name << ' ' << p->Tel << ' ' << p->Phone << endl;
112. p = p->next;
113. }
114. }
115. *//6.显示*
116. void show() {
117. cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
118. cout << "\*\*1----查询\*\*\*\*" << endl;
119. cout << "\*\*2----修改\*\*\*\*" << endl;
120. cout << "\*\*3----添加\*\*\*\*" << endl;
121. cout << "\*\*4----删除\*\*\*\*" << endl;
122. cout << "\*\*5----打印\*\*\*\*" << endl;
123. cout << "\*\*0----退出！\*\*" << endl;
124. cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
125. }
126. int main() {
127. Sta s;
128. InitList(s);
129. int choice;
131. while(1){
132. show();
134. cout << "请输入你的选择：";
135. cin >> choice;
137. switch (choice) {
138. case 1:
139. Query(s);
140. break;
141. case 2:
142. Modify(s);
143. break;
144. case 3:
145. Add(s);
146. break;
147. case 4:
148. Del(s);
149. break;
150. case 5:
151. PrintStaff(s);
152. break;
153. case 0:
154. exit(0);
155. break;
156. default:
157. cout << "输入非法，请重新输入！" << endl;
158. break;
159. }
160. }
162. return 0;
163. }