中南大学

数据结构试验报告

题 目 实验六 查找和排序

学生姓名 张子洋

学 号 8208221223

指导老师 郑瑾

学 院 计算机学院

专业班级 计算机与通信类2212班

完成时间 2023.6

指导老师评定 签名

1. 文学研究助手（设计性实验）
2. 需求分析

需求：

从键盘读入一串整数和一个待查键，查找在该整数串中是否有这个待查键。如果有，就输出它在整数串中的位置；如果没有，输出-1；

分析：

1. 输入的形式和输入值的范围：一串整数和一个待查键
2. 输出的形式：输出int型数字
3. 程序所能达到的功能：查找在该整数串中是否有这个待查键
4. 测试数据：

* 1 2 4 5 7 8 11 14 15 19 20 51 100 1212
* 20

1. 概要设计

本程序包含2个函数：

* 主函数main（）；
* 二分查找search（）；

1. 详细设计

详细代码如下：

1. #include<stdio.h>
2. #include<stdlib.h>
3. //二分查找非递归方式
4. int HalfSearch(int a[],int low,int high,int key)
5. {
6. int mid;
7. while(low<=high)
8. {
9. mid=(low+high)/2;//二分点
10. if(a[mid]==key) return mid;
11. else if(a[mid]<key) low=mid+1;
12. else high=mid-1;
13. }
14. return -1;
15. }
16. //二分查找递归方式
17. int HalfSearch1(int a[],int low,int high,int key)
18. {
19. int mid;
20. if(low>high) return -1;//没找到，递归结束
21. else{
22. mid=(low+high)/2;
23. if(a[mid]==key) return mid;//找到了，递归结束
24. else if(a[mid]<key) return HalfSearch(a,mid+1,high,key);//右递归
25. else return HalfSearch(a,low,mid-1,key);//左递归
26. }
27. }
28. int main(void)
29. {
30. int a[10]={12,22,34,65,76,78,89,90,121,145};
31. int n;
32. int key;
33. printf("enter data to key:\n");
34. scanf("%5d",&key);
35. n=HalfSearch1(a,0,9,key);//也可以使用上一个函数
36. if(n==-1) printf("not found!\n");//没找到
37. else printf("位置在：a[%d]",n);
38. return 0;
39. }

4.调试分析

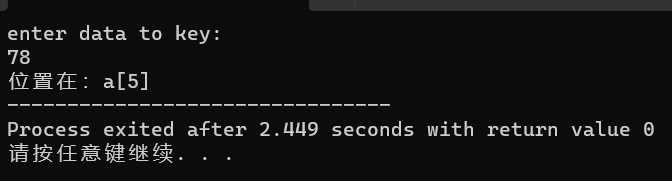
1. 采用IDE中自带的调试功能进行调试，手动添加断点和查看程序。
2. 对设计和编码的讨论和分析。该程序实现了顺序栈的操作。分析程序代码的质量，主要从以下几个方面考虑。 

* 正确性。在一定的数据范围内，该程序能实现所需功能，所以正确性是没有问题的。
* 健壮性。在一定的数据输入范围内，该程序能较好的实现链表的操作。但是如果输入数 据非法，该程序还是可能会产生一些预想不到的输出结构，或是不做任何处理。所以， 该程序的健壮性有待进一步的提高。要综合考虑一些情况，当输入有误时，应返回一个 表示错误的值，并中止程序的执行，以便在更高的抽象层次上进行处理。

1. 使用说明

按照屏幕提示，输入一串整数和一个待查数，按下ENTER键即可得到结果。

1. 测试程序的运行结果



1. 心得体会

本次实验比较简单，算法上也好理解。

附录：源程序文件清单

各程序源代码文件随本实验报告电子版一起打包，存放在文件夹二分查找算法中。

文件清单如下：

二分查找算法.c…………………………………………二分查找算法的实现

1. 简单个人电话号码查询系统（综合性实验）
2. 需求分析

需求：

人们在日常生活中经常要查找某个人或某个单位的电话号码，本实验将实现一个简单的个人电话号码查询系统，根据用户输入的信息（如姓名等）进行快速查询。

基本要求

(1) 在外存上，用文件保存电话号码信息。

(2) 在内存中，设计数据结构存储电话号码信息。

(3) 提供查询功能：根据姓名实现快速查询。

(4) 提供其他维护功能，如插入、删除、修改等。

分析：

1. 输入的形式和输入值的范围：根据提示输入int型数字选择指令，再根据屏幕提示输入对应数据。
2. 输出的形式：输出对应指令结果。
3. 程序所能达到的功能：
4. 快速查询
5. 插入
6. 删除
7. 修改
8. 测试数据：

* 1
* 赵楷鹏
* 2
* 小明
* 11155523632
* 254687956
* 1321331132
* 1
* 小明
* 3
* 小明
* 1
* 小明
* 4
* 赵楷鹏
* 11111111111
* 22222222
* 33333333333333333
* 1
* 赵楷鹏
* 5

1. 概要设计
2. 本程序包含3个函数：

* 主函数main()；
* 将string转化为unsigned int类型 ELFhash ()；;
* hash查找found ()；

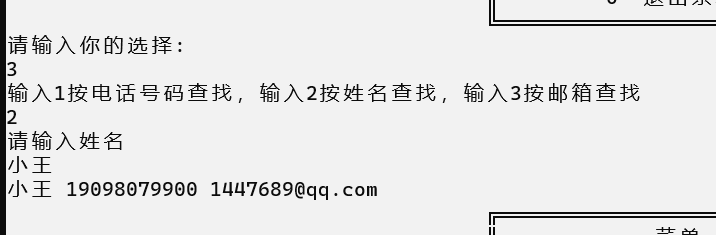
1. 详细设计
2. #include<stdio.h>
3. #include<stdlib.h>
4. #include<string.h>
5. /\*12334567890
6. 小李
7. 12345678@qq.com
8. 19098079900
9. 小王
10. 1447689@qq.com\*/
11. #include<iostream>
12. int d[50];/\*再散列\*/
13. struct list
14. {
15. char a[12];//电话号码
16. char name[15];//名字
17. char add[20];//邮箱
18. int f=0;//标记是否存在
19. };
20. struct sqstack
21. {
22. struct list \*base;
23. int i;
24. };
25. struct sqstack S;
26. void store(char \*a,char \*name,char \*add)
27. {
28. int key;
29. key=int(a[0])+int(a[3])+int(a[7]);/\*以电话号码的第1,4,8位作为关键字构造哈希函数\*/
30. S.i=key%20;
31. int j=1;
32. while(true)
33. {
34. if((S.base+S.i)->f==0)
35. {
36. strcpy((S.base+S.i)->a,a);
37. strcpy((S.base+S.i)->name,name);
38. strcpy((S.base+S.i)->add,add);
39. (S.base+S.i)->f=1;
40. break;
41. }
42. S.i=(key%20+d[j])%20;
43. j++;//解决哈希冲突
44. }
46. }
47. void shuru()
48. {
49. void interface();
50. printf("请输入：\n例如：\n15556931615\n小王\n12345678@qq.com\n输入0结束\n\n");
51. char a[12];
52. char name[15];
53. char add[15];
54. while(true)
55. {
56. scanf("%s",a);
57. if(a[0]=='0')
58. break;
59. scanf("%s",name);
60. scanf("%s",add);
61. printf("%s已保存\n",name);
62. store(a,name,add);/\*将输入保存到哈希表\*/
63. }
64. interface();
65. }
66. void print()
67. {
68. void interface();
69. int i;
70. printf("姓名 电话号码 邮箱\n");
71. for(i=0;i<20;i++)
72. {
73. if((S.base+i)->f==1)
74. {
75. printf("%s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
76. }
77. }
78. interface();
79. }
80. void search()
81. {
82. void interface();
83. int i;
84. int ff=0;
85. int b;
86. char a[15];
87. printf("输入1按电话号码查找，输入2按姓名查找，输入3按邮箱查找\n");
88. scanf("%d",&b);
89. switch(b)
90. {
91. case 1:printf("请输入电话号码\n");
92. scanf("%s",a);
93. for(i=0;i<20;i++)
94. if(strcmp(a,(S.base+i)->a)==0)
95. {
96. printf("%s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
97. ff=1;
98. }
99. if(ff==0)
100. printf("找不到该用户\n");
101. break;
102. case 2:printf("请输入姓名\n");
103. scanf("%s",a);
104. for(i=0;i<20;i++)
105. if(strcmp(a,(S.base+i)->name)==0)
106. {
107. printf("%s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
108. ff=1;
109. }
110. if(ff==0)
111. printf("找不到该用户\n");
112. break;
113. case 3:printf("请输入邮箱\n");
114. scanf("%s",a);
115. for(i=0;i<20;i++)
116. if(strcmp(a,(S.base+i)->add)==0)
117. {
118. printf("%s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
119. ff=1;
120. }
121. if(ff==0)
122. printf("找不到该用户\n");
123. break;
124. }
125. interface();
126. }
127. void del()
128. {
129. void interface();
130. int i;
131. int ff=0;
132. int b;
133. char a[15];
134. printf("输入1按电话号码删除，输入2按姓名删除，输入3按邮箱删除\n");
135. scanf("%d",&b);
136. switch(b)
137. {
138. case 1:printf("请输入电话号码\n");
139. scanf("%s",a);
140. for(i=0;i<20;i++)
141. if(strcmp(a,(S.base+i)->a)==0)
142. {
143. (S.base+i)->f=0;
144. printf("已删除： %s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
145. ff=1;
146. }
147. if(ff==0)
148. printf("找不到该用户\n");
149. break;
150. case 2:printf("请输入姓名\n");
151. scanf("%s",a);
152. for(i=0;i<20;i++)
153. if(strcmp(a,(S.base+i)->name)==0)
154. {
155. (S.base+i)->f=0;
156. printf("已删除： %s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
157. ff=1;
158. }
159. if(ff==0)
160. printf("找不到该用户\n");
161. break;
162. case 3:printf("请输入邮箱\n");
163. scanf("%s",a);
164. for(i=0;i<20;i++)
165. if(strcmp(a,(S.base+i)->add)==0)
166. {
167. (S.base+i)->f=0;
168. printf("已删除： %s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
169. ff=1;
170. }
171. if(ff==0)
172. printf("找不到该用户\n");
173. break;
174. }
175. interface();
176. }
177. void change()
178. {
179. void interface();
180. int i;
181. int ff=0;
182. int b;
183. char a[15];
184. printf("请输入姓名\n");
185. scanf("%s",a);
186. for(i=0;i<20;i++)
187. if(strcmp(a,(S.base+i)->name)==0)
188. {
189. printf("您要修改的是： %s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
190. printf("请输入新信息\n");
191. scanf("%s",(S.base+i)->a);
192. scanf("%s",(S.base+i)->name);
193. scanf("%s",(S.base+i)->add);
194. printf("已修改成： %s %s %s\n",(S.base+i)->name,(S.base+i)->a,(S.base+i)->add);
195. ff=1;
196. }
197. if(ff==0)
198. printf("找不到该用户\n");
200. interface();
201. }
202. void write()
203. {
204. void interface();
205. int i=0;
206. FILE \*fp;
207. if((fp=fopen("通讯录.txt","wb"))==NULL)
208. {
209. printf("open file error\n");
210. exit(1);
211. }
212. for(i=0;i<=20;i++)
213. {
214. int ch=32;
215. if((S.base+i)->f==1)
216. {
217. fprintf(fp,"%s",(S.base+i)->name);fputc(ch,fp);
218. fprintf(fp,"%s",(S.base+i)->a);fputc(ch,fp);
219. ch=10;
220. fprintf(fp,"%s",(S.base+i)->add);fputc(ch,fp);
221. }
222. }
223. fclose(fp);
224. interface();
225. }
226. void interface()
227. {
228. int i;
229. printf("\t\t\t\t\t╔════════════════════════════════╗\n");
230. printf("\t\t\t\t\t║ 菜单 ║\n");
231. printf("\t\t\t\t\t║════════════════════════════════║\n");
232. printf("\t\t\t\t\t║ 1--输入通讯录 ║\n");
233. printf("\t\t\t\t\t║════════════════════════════════║\n");
234. printf("\t\t\t\t\t║ 2--显示通讯录 ║\n");
235. printf("\t\t\t\t\t║════════════════════════════════║\n");
236. printf("\t\t\t\t\t║ 3--查找联系人 ║\n");
237. printf("\t\t\t\t\t║════════════════════════════════║\n");
238. printf("\t\t\t\t\t║ 4--删除联系人 ║\n");
239. printf("\t\t\t\t\t║════════════════════════════════║\n");
240. printf("\t\t\t\t\t║ 5--修改联系人信息 ║\n");
241. printf("\t\t\t\t\t║════════════════════════════════║\n");
242. printf("\t\t\t\t\t║ 6--保存到文件 ║\n");
243. printf("\t\t\t\t\t║════════════════════════════════║\n");
244. printf("\t\t\t\t\t║ 0--退出系统 ║\n");
245. printf("\t\t\t\t\t╚════════════════════════════════╝\n");
246. printf("请输入你的选择:\n");
247. scanf("%d",&i);
248. switch(i)
249. {
250. case 0:return ;break;
251. case 1:shuru();break;
252. case 2:print();break;
253. case 3:search();break;
254. case 4:del();break;
255. case 5:change();break;
256. case 6:write();break;
257. };
258. }
259. int main()
260. {
261. system("color F0"); //可以写成 red 调出颜色组
262. S.base=(struct list \*)malloc(20\*sizeof(struct list));
263. printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎使用通讯录查询系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");
264. system("date /T");
265. system("TIME /T");
266. int i;
267. for(i=1;i<25;i++)
268. d[2\*i]=-1\*i\*i;
269. for(i=1;i<25;i++)/\*构造二次再散列\*/
270. d[i+i-1]=i\*i;
271. interface();
272. }调试分析
273. 采用IDE中自带的调试功能进行调试，手动添加断点和查看程序。
274. 对设计和编码的讨论和分析。该程序实现了城市链表的操作。分析程序代码的质量，主要从以下几个方面考虑。 

* 正确性。在一定的数据范围内，该程序能实现所需功能，所以正确性是没有问题的。
* 健壮性。在一定的数据输入范围内，该程序能较好的实现链表的操作。但是如果输入数 据非法，该程序还是可能会产生一些预想不到的输出结构，或是不做任何处理。所以， 该程序的健壮性有待进一步的提高。要综合考虑一些情况，当输入有误时，应返回一个 表示错误的值，并中止程序的执行，以便在更高的抽象层次上进行处理。

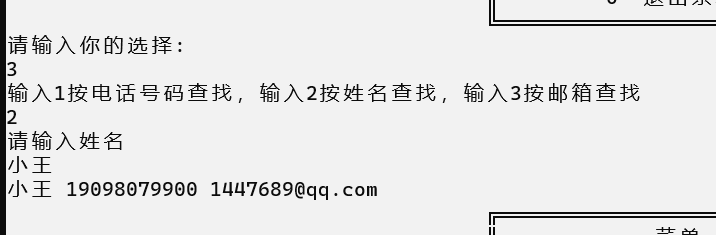
1. 使用说明

按照屏幕提示，将数据放入对应文件中，根据屏幕提示进行输入，即可得到想要的结果。

1. 测试程序的运行结果









1. 心得体会

本次实验中，我也遇到了很多挑战和问题，例如如何设计合适的数据结构、如何进行错误处理、如何提升系统的性能等。这些问题需要不断思考和解决，同时也需要不断地学习新的知识和技能。我在开发简单个人电话号码查询系统的过程中，深刻体悟到了软件开发过程的重要性和复杂性。

附录：源程序文件清单

各程序源代码文件随本实验报告电子版一起打包，存放在文件夹简单个人电话号码查询系统中。

文件清单如下：

简单个人电话号码查询系统.c…………………………………………简单个人电话号码查询系统的实现