

# 第五次实验：实现电话号码表的增删查改功能

该次实验是在第四次的实验的基础上进行功能的添加，实现可以添加联系人、删除联系人、寻找联系人和修改联系人的信息。

- 在本次实验中添加联系人的时候是默认添加到电话号码表的末尾；
- 寻找联系人就是对电话号码表进行一次简单的遍历；
- 删除联系人的话，本次实验设计思路是将最后一个联系人放在要删除的联系人的位置；
- 修改联系人的话，就是寻找联系人和添加联系人功能的结合；

## 第五次实验：实现电话号码表的增删查改功能

### 一. 具体功能实现：

- 1.0 电话号码表在内存中的结构
- 1.1 实验程序的整体架构
- 1.2 查询联系人功能
- 1.3 添加联系人的功能
- 1.4 删除联系人的功能实现
- 1.5 修改联系人信息功能实现
- 1.6 输出整个电话表的信息

### 附录1:输出结果

- 添加联系人：
- 输出所有信息：
- 寻找联系人：
- 修改联系人信息：
- 删除联系人信息：

### 附表2：具体代码实现

## 一. 具体功能实现：

### 1.0 电话号码表在内存中的结构

电话号码表中联系人的相关信息在内存中的结构如下所示：

|            |                        |               |
|------------|------------------------|---------------|
| 076C： 0018 | ..kirito ..17698266260 | ----->38Bytes |
| 076C： 003E | ..Sam ..12345678909    | ----->38Bytes |
| 076C： 0064 | ..Kite ..12345343434   | ----->38Bytes |
| 076C： 008A | ..Siri ..12345343434   | ----->38Bytes |

代码中定义：

```
1 | tel_tab db 80 dup(20,?,20 dup(' '),32,12,?,12 dup(0),32);存储人名和电话号码
```

```

..kirito
.
.17698266260. ..
Sam.t
..123456789
09. ..Kite.
..345
67890987. ..

```

## 1.1 实验程序的整体架构

主函数所做的事情就是一直循环输出功能列表，用户选择相应的功能之后开始执行相关的操作。

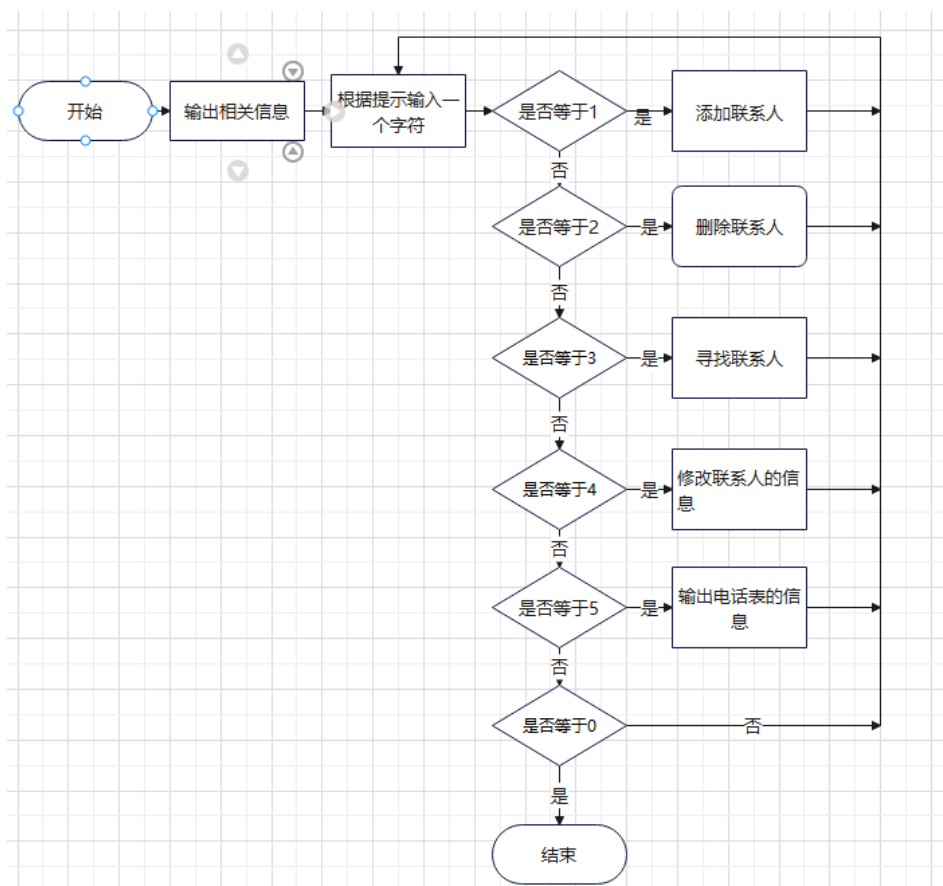
该部分很简单：就是首先屏幕会输出一个列表。上面是功能介绍，本次实验的具体功能有：

```

-----menu-----
Add a people to the table-----1
Delete a people from table-----2
Search a people from table-----3
Change detail of the people-----4
Show all detail of the book-----5
return-----0
Please input num to choose:

```

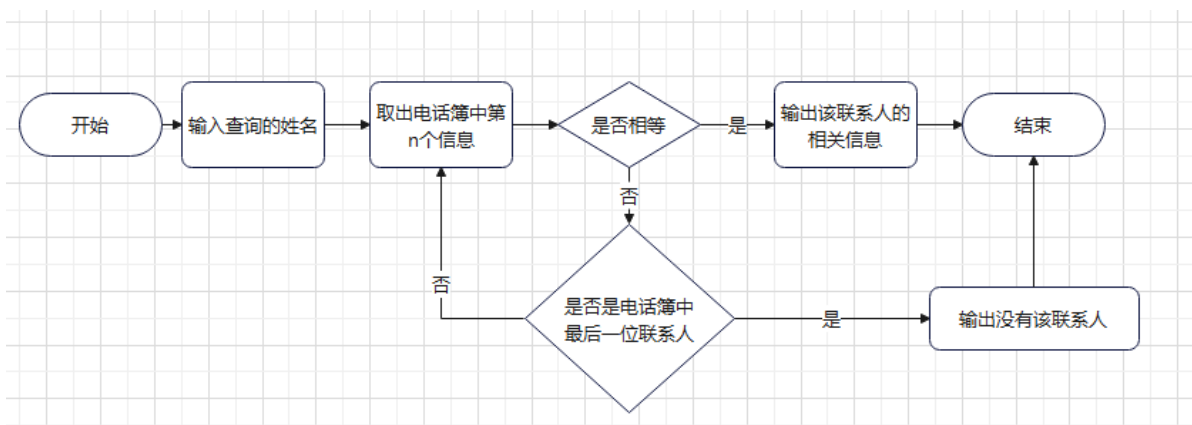
每一次会提示用户输入一个字符，接着会根据输入的具体的字符进行判断而去执行不同的子程序。



## 1.2 查询联系人功能

首先会提示用户输入要查询的联系人的姓名，本次实验中只提供了根据姓名去寻找指定联系人的功能。

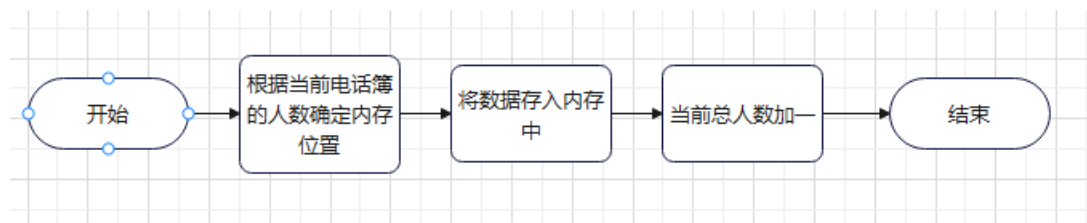
接着遍历即可。



## 1.3 添加联系人的功能

该实验中，其中有一个全局变量来存储电话簿中存储的联系人的总数，很多功能的都是根据这个变量来实现的，除此之外还有一个变量来存储该电话簿最多存储的联系人的总数。

我们添加联系人的时候，首先先确定我们即将添加的联系人准备存储的内存位置，然后将数据存入其中，最后将当前电话簿的联系人的总数加一。

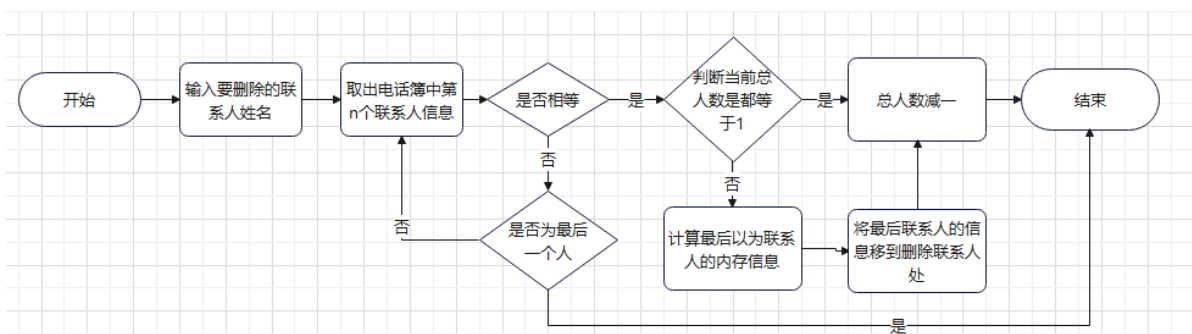


## 1.4 删除联系人的功能实现

该功能实现的时候借助于前面已经造好的轮子，即联系人寻找功能。首先会提示输入要删除的联系人的姓名，接着会在电话簿中遍历寻找该联系人，如果未找到的话，说明该电话簿中不存在该联系人，结束程序。如果找到的话，计算最后一位联系人的内存位置，并且将该联系人的数据放在我们要删除的联系人的内存位置，替换掉，当前总人数减一即可。

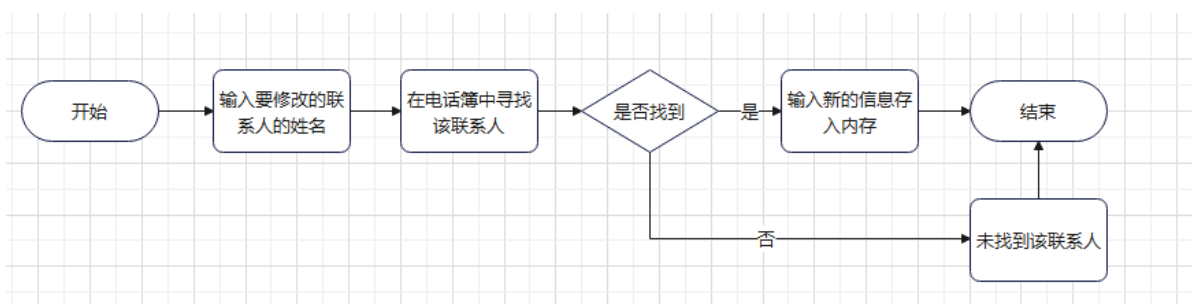
注意一点就是，如果联系人总人数为1的话，我们可以直接将当前总人数减一即可完成删除联系人的目的。

相关流程图如下：



## 1.5 修改联系人信息功能实现

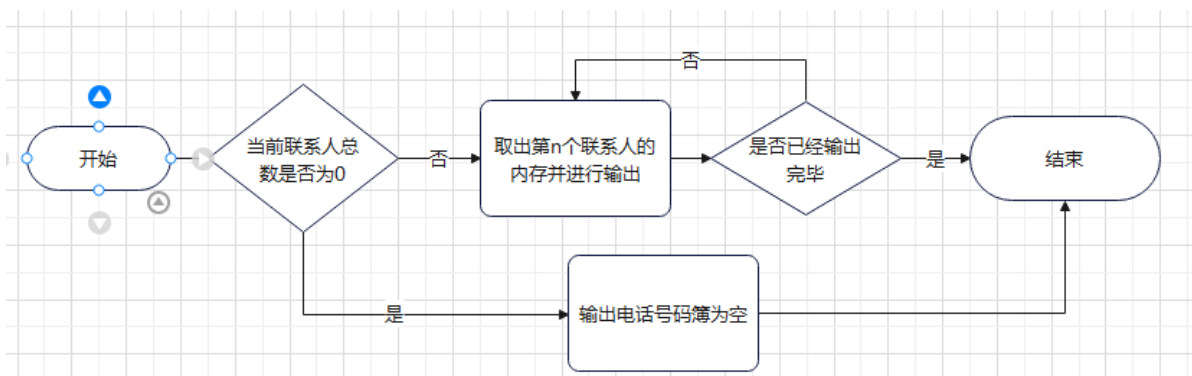
该处功能的实现就是查询联系任何添加联系人的功能的结合，较为简单，暂不展开阐述。



## 1.6 输出整个电话表的信息

根据当前电话簿中的总人数来进行输出和每一个联系人的内存位置的查找。

如果当前人数为0的话，输出电话簿为空，否则就先取出第一个联系人的内存地址，将其中的信息一个一个字符进行输出，电话号码和姓名直接用空格隔开。



## 附录1:输出结果

添加联系人：

```

-----menu-----
Add a people to the table-----1
Delete a people from table-----2
Search a people from table-----3
Change detail of the people-----4
Show all detail of the book-----5
return-----0
Please input num to choose: 1
Please input name: kirito
Please input the phone_num: 17698266260
  
```

输出所有信息：

```
Please input num to choose: 5

The telephone book is below:
kirito 2222222222
sam 12345678909
kir 12345676564
asdda 123412312
zxcxzc 312412412
adscdsa 123443124
dascdsadc 1234124321
```

寻找联系人:

---

```
-----menu-----
Add a people to the table-----1
Delete a people from table-----2
Search a people from table-----3
Change detail of the people-----4
Show all detail of the book-----5
return-----0
Please input num to choose: 3
Please input name:kirito
kirito 17698266260
```

修改联系人信息:

---

```
return-----0
Please input num to choose: 4

Please input the name you want to change:kirito
Please input the number you want:2222222222
Change the detail of the man successfully!
```

删除联系人信息:

---

```

1 Add a people to the table-----1
2 Delete a people from table-----2
3 Search a people from table-----3
4 Change detail of the people-----4
5 Show all detail of the book-----5
6 return-----0
Please input num to choose: 2
Please input the name to delete: kirito

```

可以从下图看出该联系人的信息已经删除。

```

Please input num to choose: 5

The telephone book is below:
scdas 2121
sam 12345678909
kir 12345676564
asdda 123412312
zxcxzc 312412412
adscdsa 123443124
dascdsadc 1234124321
dasdas 3213212

```

## 附表2：具体代码实现

```

1  assume ds:data,cs:code,ss:stacks
2  data segment
3      sum dw 0;记录当前电话簿中有多少联系人
4      namemax db 20
5      _length db ?
6      _name db 20 dup(' ')
7      tel_tab db 80 dup(20,?,20 dup(' '),32,12,?,12 dup(0),32);存储人名和电话号
8
9      menu db 13,10,"-----menu-----",13,10,'$'
10     menu1 db "Add a people to the table-----1",13,10,'$'
11     menu2 db "Delete a people from table-----2",13,10,'$'
12     menu3 db "Search a people from table-----3",13,10,'$'
13     menu6 db "Change detail of the people-----4",13,10,'$'
14     menu7 db "Show all detail of the book-----5",13,10,'$'
15     menu4 db "return-----0",13,10,'$'
16     menu5 db "Please input num to choose: ",'$'
17     temp db "Please input name: ",'$'
18     tel_num db 13,10,"Please input the phone_num: ",'$'
19     CRLF db 13,10,'$'
20     method db "Please input name: ",'$'
21     method_0 db "Please input the name to delete: ",'$'
22     temp6 db 13,10,"The table is empty!",'$'

```

```

22     temp7 db 13,10,"Failed to find the people!",$'
23     temp8 db 13,10,"The telephone book is below:",13,10,$'
24     temp9 db 13,10,"Please input the name you want to change:",'$'
25     temp10 db 13,10,"Please input the number you want:",'$'
26     temp11 db 13,10,"Change the detail of the man successfully!",13,10,$'
27
28 data ends
29 stacks segment stack
30     dw 20 dup(?)
31 stacks ends
32
33 code segment
34 start:
35     mov ax,data
36     mov ds,ax
37     mov ax,stacks
38     mov ss,ax
39 main_loop:
40     call show_men
41     mov ah,1
42     int 21H;从键盘输入数字选择
43     push ax
44     mov dx,offset CRLF
45     mov ah,09h
46     int 21H
47     pop ax
48     xor ah,ah
49     cmp ax,0030h
50     je return;如果输入的是0，则结束程序
51     cmp ax,0031h
52     je loop1;如果输入的是1, 2, 3, 4, 5，则进入子程序
53     cmp ax,0032h
54     je loop2
55     cmp ax,0033h
56     je loop3
57     cmp ax,0034h
58     je loop4
59     cmp ax,0035h
60     je loop5
61     jmp main_loop
62
63 return:
64     mov ah,4CH
65     int 21H
66 loop1:
67     call add_man
68     jmp main_loop
69 loop2:
70     call del_man
71     jmp main_loop
72 loop3:
73     call sear_man
74     jmp main_loop
75 loop4:
76     call change

```

```

77     jmp main_loop
78 loop5:
79     call show_book
80     jmp main_loop
81
82 ;输出菜单
83 show_men proc near
84     mov dx,offset menu
85     mov ah,09h
86     int 21h
87     mov dx,offset menu1
88     mov ah,09h
89     int 21h
90     mov dx,offset menu2
91     mov ah,09h
92     int 21h
93     mov dx,offset menu3
94     mov ah,09h
95     int 21h
96     mov dx,offset menu6
97     mov ah,09h
98     int 21h
99     mov dx,offset menu7
100    mov ah,09h
101    int 21h
102    mov dx,offset menu4
103    mov ah,09h
104    int 21h
105    mov dx,offset menu5
106    mov ah,09h
107    int 21h
108    ret
109 show_men endp
110
111 ;输出所有联系人
112 show_book proc near
113     mov dx,offset temp8
114     mov ah,09h
115     int 21h
116
117     push sum
118     cmp sum,0
119     je show_empty;如果电话簿为空的话，输出相应信息
120     mov cx,sum
121
122
123     mov si,offset tel_tab
124 show_big:
125     push cx;外层循环，遍历每一位联系人
126     add si,1
127     mov cl,ds:[si]
128     xor ch,ch
129     add si,1
130     push si
131 show_name:;输出姓名

```



```

132     mov dl,ds:[si]
133     mov ah,02h
134     int 21h
135     inc si
136     loop show_name
137
138     pop si
139     add si,22
140     mov cl,ds:[si]
141     xor ch,ch
142     add si,1
143     push si
144
145     mov dl,' '
146     mov ah,02h
147     int 21h
148 show_phonenum:;输出电话号码
149     mov dl,ds:[si]
150     mov ah,02h
151     int 21h
152     inc si
153     loop show_phonenum
154     mov dx,offset CRLF
155     mov ah,09h
156     int 21h
157
158     pop si
159     add si,13
160     pop cx
161     loop show_big
162     jmp show_ret
163
164 show_empty:
165     mov dx,offset temp6
166     mov ah,09h
167     int 21h
168 show_ret:
169     pop sum
170     ret
171 show_book endp
172
173 ;添加联系人
174 add_man proc near
175     lea dx,temp
176     mov ah,09h
177     int 21h
178     mov dx,offset tel_tab
179     push sum
180 init:;初始化该联系人存放的内存位置
181     cmp sum,0
182     je add_loop
183     add dx,38
184     sub sum,1
185     jmp init
186 add_loop:

```

```

187     pop sum
188     mov ah,0ah
189     int 21H
190     push dx
191
192     mov dx,offset tel_num
193     mov ah,09h
194     int 21H
195
196     pop dx
197     add dx,23
198     mov ah,0ah
199     int 21h
200
201     inc sum;计数器加一
202     ret
203 add_man endp
204
205 ;删除联系人
206 del_man proc near
207     push sum;暂存联系人的人数
208
209     mov dx,offset method_0
210     mov ah,09h
211     int 21H
212
213     ;输入要删除的联系人
214     mov dx,offset namemax
215     mov ah,0ah
216     int 21H
217
218     cmp sum,1
219     je dele_brige2
220
221     ;取出比较的次数
222     mov cl,_length
223     xor ch,ch
224     ;初始化电话簿的地址
225     mov si,offset tel_tab
226     add si,2
227     mov di,seg _name
228     mov es,di
229     mov di,offset _name
230     mov bx,0;计数器，记录我们要删除的联系人的位置
231     cmp sum,0
232     je dele_brige;如果sum一开始就为0，即为空的，肯定找不到
233 dele_loop4:
234     cmp sum,0;此时肯定不会满足条件，所以开始寻找
235     je dele_brige1;由于跳转的位置距离此处太远所以需要跳板
236     inc bx
237 dele_cmp:
238     push bx
239     mov al,ds:[si]
240     mov bl,es:[di]
241     cmp al,bl

```

```

242     pop bx
243     jne dele_nomatch
244     inc si
245     inc di
246     loop dele_cmp
247     jmp dele_suce
248 dele_nomatch:
249     add si,38
250     mov di,offset _name
251     sub sum,1
252     jmp dele_loop4
253 dele_brige:
254     jmp dele_empty
255 dele_brige1:
256     jmp dele_fail
257 dele_brige2:
258     jmp dele_only
259 ;查找成功
260 ;找到并初始化我们要删除的联系人的位置
261 dele_suce:
262     mov cx,bx
263     sub cx,1
264     mov si,offset tel_tab
265 dele_init:
266     cmp cx,0
267     je dele_init_
268     add si,38
269     loop dele_init
270
271 dele_init_:
272 ;初始化最后一位联系人的位置
273     pop cx;取出当前联系人的总人数
274     push cx
275
276     mov di,seg _name
277     mov es,di
278     mov di,offset tel_tab
279     sub cx,1;如果当前总人数为1的话，就不需要循环初始化
280     cmp cx,0
281     je _exchange
282 dele_loop:
283     add di,38
284     loop dele_loop
285     push si
286     push di
287 ;交换最后一位联系人和我们要删除的联系人的姓名位置
288 _exchange:
289     mov cx,22
290 _xchange:
291     mov al,es:[di]
292     mov byte ptr ds:[si],al
293     inc di
294     inc si
295     loop _xchange
296 ;交换相应联系人电话号码

```

```

297     pop di
298     pop si
299     add si,23
300     add di,23
301     mov cx,14
302     _xchange_num:
303         mov al,es:[di]
304         mov byte ptr ds:[si],al
305         inc di
306         inc si
307         loop _xchange_num
308     dele_exit:
309         pop sum
310     ;删除一个联系人之后，总人数减少
311         sub sum,1
312         ret
313     dele_only:
314         mov dx,offset temp11
315         mov ah,09h
316         int 21H
317         pop sum
318         sub sum,1
319         ret
320     ;查找失败
321     dele_fail:
322         mov dx,offset temp7
323         mov ah,09h
324         int 21H
325         pop sum
326         ret
327     ;表为空
328     dele_empty:
329         mov dx,offset temp6
330         mov ah,09h
331         int 21H
332         pop sum
333         ret
334     del_man endp
335
336     ;寻找联系人
337     sear_man proc near
338         push sum;暂存联系人的人数
339         mov dx,offset method
340         mov ah,09h
341         int 21H
342
343         ;输入要寻找的联系人
344         mov dx,offset namemax
345         mov ah,0ah
346         int 21H
347
348         ;取出比较的次数
349         mov cl,_length
350         xor ch,ch
351         ;初始化电话簿的地址

```

```

352     mov si,offset tel_tab
353     add si,2
354     mov di,seg _name
355     mov es,di
356     mov di,offset _name
357     mov bx,0
358     cmp sum,0
359     je sear_empty;如果sum一开始就为0，即为空的，肯定找不到
360 sear_loop4:
361     cmp sum,0;此时肯定不会满足条件，所以开始寻找
362     je sear_fail
363     inc bx
364 sear_cmp:
365     push bx
366     mov al,ds:[si]
367     mov bl,es:[di]
368     cmp al,bl
369     pop bx
370     jne nomatch
371     inc si
372     inc di
373     loop sear_cmp
374     jmp sear_suce
375 nomatch:
376     add si,38
377     mov di,offset _name
378     sub sum,1
379     jmp sear_loop4
380
381 ;查找成功
382 sear_suce:
383     mov cx,bx
384     sub cx,1
385     mov si,offset tel_tab
386 _init:
387     cmp cx,0
388     je _output
389     add si,38
390     loop _init
391
392
393 _output:
394     add si,25
395     push si
396     mov si,offset _name
397     mov cl,_length
398     xor ch,ch
399
400     mov dx,offset CRLF
401     mov ah,09h
402     int 21h
403
404 _out_name:
405     mov dl,ds:[si]
406     mov ah,02h

```

```

407     int 21H
408     inc si
409     loop _out_name
410
411     pop si
412     mov cx,11
413     mov dl,' '
414     mov ah,02h
415     int 21H
416 _out_number:
417     mov dl,ds:[si]
418     mov ah,02h
419     int 21H
420     inc si
421     loop _out_number
422 sear_exit:
423     pop sum
424     ret
425 ;查找失败
426 sear_fail:
427     mov dx,offset temp7
428     mov ah,09h
429     int 21H
430     jmp sear_exit
431 ;表为空
432 sear_empty:
433     mov dx,offset temp6
434     mov ah,09h
435     int 21H
436     jmp sear_exit
437 sear_man endp
438
439 ;修改联系人信息
440 change proc near
441     push sum;暂存联系人的人数
442     mov dx,offset temp9
443     mov ah,09h
444     int 21H
445
446     ;输入要寻找的联系人
447     mov dx,offset namemax
448     mov ah,0ah
449     int 21H
450
451     ;取出比较的次数
452     mov cl,_length
453     xor ch,ch
454     ;初始化电话簿的地址
455     mov si,offset tel_tab
456     add si,2
457     mov di,seg _name
458     mov es,di
459     mov di,offset _name
460     mov bx,0
461     cmp sum,0

```

```

462     je change_empty;如果sum一开始就为0，即为空的，肯定找不到
463 change_loop4:
464     cmp sum,0;此时肯定不会满足条件，所以开始寻找
465     je change_fail
466     inc bx
467 change_cmp:
468     push bx
469     mov al,ds:[si]
470     mov bl,es:[di]
471     cmp al,bl
472     pop bx
473     jne change_nomatch
474     inc si
475     inc di
476     loop change_cmp
477     jmp change_suce
478 change_nomatch:
479     add si,38
480     mov di,offset _name
481     sub sum,1
482     jmp change_loop4
483
484 ;查找成功
485 ;找到并初始化我们要修改的联系人的位置
486 change_suce:
487     mov cx,bx
488     sub cx,1
489     mov si,offset tel_tab
490 change_init:
491     cmp cx,0
492     je change_main
493     add si,38
494     loop change_init
495     push si;si保存的是我们要修改的联系人的内存位置
496
497 change_main:
498     mov dx,offset temp10
499     mov ah,09h
500     int 21h
501
502     mov dx,si
503     add dx,23
504     mov ah,0ah
505     int 21h
506
507     mov dx,offset temp11
508     mov ah,09h
509     int 21h
510
511 change_exit:
512     pop sum
513     ret
514 ;查找失败
515 change_fail:
516     mov dx,offset temp7

```

```
517     mov ah,09h
518     int 21h
519     jmp change_exit
520 ;表为空
521 change_empty:
522     mov dx,offset temp6
523     mov ah,09h
524     int 21h
525     jmp change_exit
526
527 change endp
528 code ends
529 end start
```