```cpp
# include <graphics.h>      // 引用图形库头文件
# include <conio.h>
# include<stdio.h>
# include<stdlib.h>
# include<math.h>
# include<Windows.h>
# include<queue>

# define width 1280         /*定义页面宽度*/
# define height 660         /*定义页面高度*/
# define buttleNum 4        /*定义按钮数量*/
# define Frame 100          /*定义边框宽度*/
# define Spacing 30         /*定义按钮间间距*/
# define Buttom 60          /*定义到底部的距离*/
# define MenuNum 5          /*定义查询菜单数量*/
# define relationMenuNum 6  /*定义关系菜单的数量*/

struct info
{
    //个人信息
    char name[20];
    int generation;      //第几代
    char data_birth[9];  //例如20181205
    char place_birth[20];//出生地
    char data_death[9];  //逝世日期,未逝世则记为00000000
    char sex[9];             //0指女性，1为男性
    char profession[15]; //职业
    char spouse_name[20];//配偶姓名
    char parent_name[20];//双亲姓名
};

struct person
{
    info record;    //族谱个人节点
    person *child;  //指向长子
    person *brother;//指向一个兄弟
};

typedef struct stack {
    person *data[100];
    int tag[100];
    int top;
}seqstack;
void push(seqstack *s, person *t)//压栈
{
    s->data[s->top] = t;
    s->top++;
}
person *  pop(seqstack *s)//出栈
{
    if (s->top != 0)
    {
        s->top--;
        return s->data[s->top];
    }
    else
```

```
57          return NULL;
58  }
59
60
61  /*全局变量*/
62  int buttleWidth = ((width - 200) / buttleNum) - Spacing;   /*定义按钮大小*/
63  int buttleHeight = 40;
64  int MenuWidth = ((width - 200) / MenuNum) - Spacing;   /*定义菜单按钮大小*/
65  int MenuHeight = 30;   /*定义菜单按钮高度*/
66  int MouseFindNum = 0;   /*记录查找单击次数*/
67  int MouseAddNum = 0;   /*记录添加按钮的单击次数*/
68  int MouseTreeNum = 0;   /*定义查看单机次数*/
69  int MenuSpacing = 15;   /*初始化菜单间间隔*/
70
71  /*添加，查询相关全局变量*/
72
73  char name[100];   /*定义用户输入的用来查询的姓名*/
74  char year[100];   /*定义用户输入的用来查询的年月日*/
75  char finddeath[100];
76  char findprofe[100];
77  char findspouse[100];
78  char gener[20];
79  char getfather[20];
80  char addName[20];
81  int addGeneration;          /*第几代*/
82  char addData_birth[20];     /*生日*/
83  char addPlace_birth[20];    /*出生地*/
84  char addSex[20];                 /*0 女性 1 男性*/
85  char addProfession[20];     /*职业*/
86  char addSpouse_name[20];    /*配偶姓名*/
87  char addParent_name[20];    /*双亲姓名*/
88  char getInfo[100];    /*获取用户输入的信息*/
89  char getGeneration[20];    /*获取用户输入的代数*/
90  info *newpeople = NULL;
91  int relationMenuWidth = (width - 2 * Frame) / 3;   /*定义关系菜单宽度*/
92  int relationMenuHeight = 400 / relationMenuNum - 6;   /*定义关系菜单高度*/
93  char findfather[30];
94  bool ifgetName;
95
96  /*以下全局变量用来定义树的左右上下间隔，调用位置：imctree()  */
97
98  int Left = 50;   /*定义画树时左右间距*/
99  int Top = 10;   /*定义画树时上下间距*/
100 int SuoJin = 1;   /*定义每次缩进的倍数*/
101
102 /*以下全局变量用来拼接输出时的字符串，调用位置：display() */
103
104 /*创建*/
105 person *create_family(FILE *fp)
106 {
107     person *t = (person*)malloc(sizeof(person));
108     fscanf(fp, "%s", t->record.name);     //姓名后的标记为空格' '
109     if (t->record.name[0] == '#')          //姓名为#时，建立空树
110         t = NULL;
111     else
112     {
```

```
113         fscanf(fp, "%d%s%s%s%s%s%s%s", &t->record.generation,
114             t->record.data_birth, t->record.place_birth, t->record.data_death,
115             &t->record.sex, t->record.profession, t->record.spouse_name,
116             t->record.parent_name);
117         t->child = create_family(fp);
118         t->brother = create_family(fp);
119     }
120     return t;
121 }
122
123 /*查找*/
124 using namespace std;
125 void display(person *g)//输出
126 {
127     fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清空输出区域*/
128     char bir[200] = "生日";
129     char na[200] = "姓名：";
130     char sx[200] = "性别：";
131     char csd[200] = "出生地：";
132     char dxxm[200] = "配偶姓名：";
133     char ds[200] = "第几代：";
134     char fqxm[200] = "父亲姓名：";
135     char zy[200] = "职业：";
136     ;
137     LOGFONT f;
138     gettextstyle(&f);                       // 获取当前字体设置
139     f.lfHeight = 25;
140     settextcolor(BLACK);
141     setbkmode(TRANSPARENT);
142     _tcscpy_s(f.lfFaceName, _T("黑体"));    // 设置字体为“黑体”
143     settextstyle(&f);
144
145     outtextxy( 400, Buttom + 20, strcat(na,g->record.name));
146
147     if(strcmp(g->record.sex,"1")==0)
148         outtextxy(400, Buttom + 50,strcat(sx,"男"));
149     else
150         outtextxy(400, Buttom + 50, strcat(sx,"女"));
151     outtextxy( 400, Buttom + 80, strcat(bir,g->record.data_birth));
152     outtextxy( 400, Buttom + 110, strcat(csd,g->record.place_birth));
153     outtextxy( 400, Buttom + 140, strcat(zy,g->record.profession));
154     outtextxy(400, Buttom + 170, strcat(dxxm,g->record.spouse_name));
155     outtextxy( 400, Buttom + 200, strcat(fqxm,g->record.parent_name));
156
157     /*printf("姓名：%s\n", g->record.name);
158     printf("性别：\n", g->record.sex);
159     printf("出生日期：%s\n", g->record.data_birth);
160     printf("出生地：%s\n", g->record.place_birth);
161     printf("职业：%s\n", g->record.profession);
162     printf("配偶姓名：%s\n", g->record.spouse_name);
163     printf("父母姓名：%s\n\n", g->record.parent_name);*/
164 }
165 person *findname(person *t, char findname[20] )//根据姓名查找
166 {
167     seqstack *s;
168     person *k;
```

```
169     k = (person *)malloc(sizeof(person));
170     k = t;
171     s = (seqstack *)malloc(sizeof(seqstack));
172     s->top = 0;
173     while ((k != NULL) || (s->top != 0))
174     {
175         while (k)
176         {
177             if (strcmp(k->record.name, findname) == 0)
178             {
179                 return k;
180             }
181             push(s, k);
182             k = k->child;
183         }
184         if (s->top > 0)
185         {
186             k = pop(s);
187             k = k->brother;
188         }
189     }
190      return NULL;
191 }
192 person *findbirth(person *t, char birthday[20])//根据生日查找
193 {
194     seqstack *s;
195     person *k;
196     k = (person *)malloc(sizeof(person));
197     k = t;
198     s = (seqstack *)malloc(sizeof(seqstack));
199     s->top = 0;
200     while ((k != NULL) || (s->top != 0))
201     {
202         while (k)
203         {
204             if (strcmp(k->record.data_birth, birthday) == 0)
205             {
206                 display(k);
207                 return k;
208             }
209             push(s, k);
210             k = k->child;
211         }
212         if (s->top > 0)
213         {
214             k = pop(s);
215             k = k->brother;
216         }
217     }
218      return NULL;
219 }
220
221 person *fdeathdata(person *t, char deathday[20])//根据逝世日期查找
222 {
223     seqstack *s;
224     person *k;
```

```
225     k = (person *)malloc(sizeof(person));
226     k = t;
227     s = (seqstack *)malloc(sizeof(seqstack));
228     s->top = 0;
229     while ((k != NULL) || (s->top != 0))
230     {
231         while (k)
232         {
233             if (strcmp(k->record.data_death, deathday) == 0)
234             {
235                 display(k);
236                 return k;
237             }
238             push(s, k);
239             k = k->child;
240         }
241         if (s->top > 0)
242         {
243             k = pop(s);
244             k = k->brother;
245         }
246     }
247     outtextxy(300, Buttom + 20, "没有找到此人信息!");
248     return NULL;
249 }
250
251 person *findpro(person *t, char findprofession[50])//根据职业查找
252 {
253     seqstack *s;
254     person *k;
255     k = (person *)malloc(sizeof(person));
256     k = t;
257     s = (seqstack *)malloc(sizeof(seqstack));
258     s->top = 0;
259     while ((k != NULL) || (s->top != 0))
260     {
261         while (k)
262         {
263             if (strcmp(k->record.profession, findprofession) == 0)
264             {
265                 display(k);
266                 return k;
267             }
268             push(s, k);
269             k = k->child;
270         }
271         if (s->top > 0)
272         {
273             k = pop(s);
274             k = k->brother;
275         }
276     }
277     outtextxy(300, Buttom + 20, "没有找到此人信息!");
278     return NULL;
279 }
280
```

```
281
282 person *findspous(person *t, char findspouse[50])//根据配偶姓名查找
283 {
284     seqstack *s;
285     person *k;
286     k = (person *)malloc(sizeof(person));
287     k = t;
288     s = (seqstack *)malloc(sizeof(seqstack));
289     s->top = 0;
290     while ((k != NULL) || (s->top != 0))
291     {
292         while (k)
293         {
294             if (strcmp(k->record.spouse_name, findspouse) == 0)
295             {
296                 display(k);
297                 return k;
298             }
299             push(s, k);
300             k = k->child;
301         }
302         if (s->top > 0)
303         {
304             k = pop(s);
305             k = k->brother;
306         }
307     }
308     //printf("没有找到此人信息\n");
309     outtextxy(300, Buttom + 20, "没有找到此人信息!");
310     return NULL;
311 }
312
313
314 void children(person *q,char dai[10])//寻找后三代任意一代的信息
315 {
316     int jianju = 80;
317     int jianju2 = 80;
318     int jianju3 = 80;
319     LOGFONT f;
320     gettextstyle(&f);                       // 获取当前字体设置
321     f.lfHeight = 25;
322     settextstyle(&f);
323     queue<person*>qq;
324     person *p = NULL, *r = NULL;
325     p = (person *)malloc(sizeof(person));
326     r = (person *)malloc(sizeof(person));
327     r = q;
328     if (!q->child)
329     {
330         RECT r = { Frame + 15, 75,width - 135,  height - 215 };
331         //fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清空输出区域
              */
332         drawtext("他，没有孩子...", &r, DT_CENTER | DT_VCENTER);
333         return;
334     }
335     else
```

```
336     {
337         if (strcmp(dai, "1")==0)
338         {
339             outtextxy(300, Buttom + 20, "他的孩子有：");
340             /*printf("%s的孩子有", r->record.name);*/
341         }
342         q = q->child;
343         if (strcmp(dai, "1")==0)
344             outtextxy(400, Buttom + 50, q->record.name);
345             /*display(q);*/
346 
347         qq.push(q);
348         while (q->brother)
349         {
350             q = q->brother;
351             qq.push(q);
352             if (strcmp(dai, "1")==0)
353                 outtextxy(400, Buttom + jianju, q->record.name);
354                 //display(q);
355             jianju += 30;
356         }
357     }
358     if (qq.front()->child)
359         p = qq.front()->child;
360     int kk = 0;
361     while (!qq.empty() && (qq.front() != p))
362     {
363         q = qq.front(); qq.pop();
364         if (q->child)
365         {
366             q = q->child;
367             qq.push(q);
368             if (strcmp(dai, "2")==0)
369             {
370                 outtextxy(400, Buttom + jianju2, q->record.name);
371                 jianju2 += 30;
372                 kk++;
373             }
374             while (q->brother)
375             {
376                 q = q->brother;
377                 qq.push(q);
378                 if (strcmp(dai, "2")==0)
379                 {
380                     //display(q);
381                     outtextxy(400, Buttom + jianju2, q->record.name);
382                     jianju2 += 30;
383                     kk++;
384                 }
385 
386             }
387         }
388     }
389     if ((kk == 0) && (strcmp(dai, "2")==0))
390     {
391         //printf("%s没有孙子\n", r->record.name);
```

```
392         RECT r = { Frame + 15, 75,width - 135,  height - 215 };
393         drawtext("他，没有孙子...", &r, DT_CENTER | DT_VCENTER);
394         return;
395     }
396 
397     int bb = 0;
398     while (!qq.empty())
399     {
400         q = qq.front();
401         qq.pop();
402         if (q->child)
403         {
404             q = q->child;
405             qq.push(q);
406             if (strcmp(dai, "3")==0)
407             {
408                 //display(q);
409                 outtextxy(400, Buttom + jianju3, q->record.name);
410                 jianju3 += 30;
411                 bb++;
412             }
413             while (q->brother)
414             {
415                 q = q->brother;
416                 qq.push(q);
417                 if (strcmp(dai, "3")==0)
418                 {
419                     //display(q);
420                     outtextxy(400, Buttom + jianju3, q->record.name);
421                     jianju3 += 30;
422                     bb++;
423                 }
424 
425             }
426         }
427     }
428     if ((bb == 0) && (strcmp(dai, "3")==0))
429     {
430         //printf("%s没有重孙子\n", r->record.name);
431         RECT r = { Frame + 15, 75,width - 135,  height - 215 };
432         drawtext("他，没有重孙子...", &r, DT_CENTER | DT_VCENTER);
433     }
434 
435 }
436 
437 void brother(person *t, person *q)//寻找兄弟信息
438 {
439     int jianju1 = 80;
440     seqstack *s; int a = 0, k = 0;
441     s = (seqstack *)malloc(sizeof(seqstack));
442     s->top = 0;
443     while ((t) || (s->top > 0))
444     {
445         while (t)
446         {
447             push(s, t);
```

```
448             s->tag[s->top - 1] = 0;
449             t = t->child;
450         }
451         while ((s->top > 0) && (s->tag[s->top - 1] == 1))
452         {
453             t = pop(s);
454             if (strcmp(t->record.name, q->record.name) == 0)
455             {
456                 if (t->brother)
457                 {
458                     /*printf("%s的兄弟为\n", q->record.name);*/
459                     outtextxy(300, Buttom + 20, "他的兄弟为：");
460                     k++;
461                 }
462                 while (t->brother)
463                 {
464                     t = t->brother;
465                     //display(t);
466                     outtextxy(400, Buttom + jianjul, t->record.name);
467                     jianjul += 30;
468                     a++;
469                 }
470                 jianjul = 50;
471                 if ((s->tag[s->top - 1] == 1) && (s->top > 0) && (k == 0))
472                 {
473                     /*printf("%s的兄弟为\n", q->record.name);*/
474                     outtextxy(300, Buttom + 20, "他的兄弟为：");
475                 }
476                 while ((s->tag[s->top - 1] == 1) && (s->top > 0))
477                 {
478                     t = pop(s);
479                     //display(t);
480                     outtextxy(400, Buttom + jianjul, t->record.name);
481                     jianjul += 30;
482                     a++;
483                 }
484                 jianjul = 50;
485                 if (a == 0)
486                     /*printf("%s无兄弟\n", q->recorde.name); */
487                     outtextxy(300, Buttom + 20, "他是独生子女...");
488                 return;
489             }
490         }
491         if (s->top > 0)
492         {
493             t = s->data[s->top - 1];
494             s->tag[s->top - 1] = 1;
495             t = t->brother;
496         }
497         else t = NULL;
498     }
499
500 }
501
502 void parents(person *t, person *q)//查找父母信息
503 {
```

```
504     seqstack *s;
505     s = (seqstack *)malloc(sizeof(seqstack));
506     s->top = 0;
507     while ((t) || (s->top > 0))
508     {
509         while (t)
510         {
511             push(s, t);
512             s->tag[s->top - 1] = 0;
513             t = t->child;
514         }
515         while ((s->top > 0) && (s->tag[s->top - 1] == 1))
516         {
517             t = pop(s);
518             if (strcmp(t->record.name, q->record.name)==0)
519             {
520                 while ((s->tag[s->top - 1] == 1) && (s->top > 0))
521                     t = pop(s);
522                 if (s->top > 0)
523                 {
524                     t = pop(s);
525                     //printf("%s的父母信息为\n", q->recorde.name);
526                     outtextxy(300, Buttom + 20, "他的父亲为：");
527                     outtextxy(400, Buttom + 80, t->record.name);
528                     //display(t);
529                     return;
530                 }
531                 else
532                     //printf("没有%s的父亲信息\n", q->record.name);
533                     outtextxy(300, Buttom + 20, "父母信息未被录入族谱！！！");
534             }
535         }
536         if (s->top > 0)
537         {
538             t = s->data[s->top - 1];
539             s->tag[s->top - 1] = 1;
540             t = t->brother;
541         }
542         else t = NULL;
543     }
544 }
545
546 void ancestor(person *t, person *q)//寻找祖先信息
547 {
548     seqstack *s; int a = 0;
549     s = (seqstack *)malloc(sizeof(seqstack));
550     s->top = 0;
551     int jianju = 80;
552     while ((t != NULL) || (s->top != 0))
553     {
554         while (t)
555         {
556             push(s, t);
557             s->tag[s->top - 1] = 0;
558             t = t->child;
559         }
```

```
560         while ((s->top > 0) && (s->tag[s->top - 1] == 1))
561         {
562             t = pop(s);
563             if (strcmp(t->record.name, q->record.name) == 0)
564             {
565                 if (s->top > 0)
566                     outtextxy(300, Buttom + 20, "他的祖先为：");
567                     //printf("%s的祖先有\n", q->record.name);
568                 while (s->top > 0)
569                 {
570                     if (s->tag[s->top - 1] != 1)
571                     {
572                         t = pop(s);
573                         outtextxy(400, Buttom + jianju, t->record.name);
574                         jianju += 30;
575                         //display(t);
576                         a++;
577                     }
578                     else t = pop(s);
579                 }
580                 return;
581             }
582         }
583         if (s->top > 0)
584         {
585             t = s->data[s->top - 1];
586             s->tag[s->top - 1] = 1;
587             t = t->brother;
588         }
589         else t = NULL;
590     }
591     if (a == 0)
592         outtextxy(300, Buttom + 20, "祖先信息未被录入族谱！！！");
593         //printf("没有%s的祖先的信息\n", q->record.name);
594 }
595
596 void children1(person *q)//寻找孩子信息
597 {
598     int jianju = 80;
599     if (!q->child)
600         outtextxy(300, Buttom + 20, "他，没有孩子...");
601     else
602     {
603         //printf("%s的孩子有\n", q->record.name);
604         outtextxy(300, Buttom + 20,  "他的孩子有：");
605         q = q->child;
606         //display(q);
607         outtextxy(400, Buttom + jianju, q->record.name);
608         jianju += 30;
609         while (q->brother)
610         {
611             q = q->brother;
612             //display(q);
613             outtextxy(400, Buttom + jianju, q->record.name);
614             jianju += 30;
615         }
```

```
616     }
617 }
618
619 /*画圈*/
620 void draw(int x, int y, char *c) {
621     circle(x, y, 10);
622     outtextxy(x - 5, y - 5, *c);   //输出元素
623 }
624 /*递归画树*/
625 void imctree(person * tree, int x, int y)
626 {
627     if (tree != NULL)
628         draw(x, y, tree->record.name);
629     if (tree->child != NULL) {
630         SuoJin++;
631         line(x - 14, y + 14, x - Left, y + 50);
632         Sleep(100);
633         imctree(tree->child, x - Left, y + 70);
634     }
635     if (tree->brother != NULL)
636     {
637         SuoJin++;
638         line(x + 14, y + 14, x + Left, y + 50);
639         Sleep(100);
640         imctree(tree->brother, x + Left, y + 70);
641     }
642
643     SuoJin--;
644
645 }
646
647 /*初始化图形界面*/
648
649 void initImg()
650 {
651
652     IMAGE imgbk;
653     loadimage(&imgbk, _T("E:/vs2019/数据结构课设/数据结构课设/10.jpg"), 1280, 660);
654     putimage(0, 0, &imgbk);//显示图片
655
656     //loadimage(NULL, _T("10.jpg"), width, height);    /*图片填充*/
657     //fillrectangle(Frame, 60, width - 100, height - 200);   /*绘制输出区域*/
658     IMAGE img;
659     loadimage(&img, _T("E:/vs2019/数据结构课设/数据结构课设/11.jpg"), 1080, 400);
660     putimage(100, 60, &img);   //显示图片
661
662 }
663
664 /*绘制按钮*/
665
666 void DrawButtle()
667 {
668     LOGFONT f;
669     gettextstyle(&f);
670     f.lfHeight = 35;
671     settextcolor(BLACK);
```

```
672     int left = Frame + Spacing / 2;
673     int top = height - 200 + Buttom;
674     int i;
675     char s0[15] = "查找";
676     char s1[15] = "插入";
677     char s2[15] = "退出";
678     char s3[15] = "查看";
679     char *s[] = { s0,s1,s2,s3};
680     for (i = 0; i < buttleNum; i++)
681     {
682         fillrectangle(left, top, left + buttleWidth, top + buttleHeight);
683         outtextxy(left + 85, top + 6, s[i]);
684         left = left + buttleWidth + Spacing;
685     }
686 }
687
688 /*绘制查询菜单*/
689
690 void queryMenu()
691 {
692     int i;
693     char s0[15] = "根据姓名查询";
694     char s1[15] = "根据出生日期";
695     char s2[15] = "根据逝世日期";
696     char s3[15] = "根据职业查询";
697     char s4[30] = "根据配偶查询";
698     char *s[] = { s0,s1,s2,s3,s4 };
699     int g;
700     LOGFONT f;
701     gettextstyle(&f);                       // 获取当前字体设置
702     f.lfHeight = 25;
703     settextstyle(&f);
704     int menuLeft = Frame + Spacing / 2;   /*定义菜单按钮初始化位置*/
705     int MenuTop = height - 2*Frame + Buttom + buttleHeight + 20;   /*菜单初始化高度*/
706
707     for (i = 0; i < MenuNum; i++)
708     {
709         fillrectangle(menuLeft, MenuTop, menuLeft + MenuWidth, MenuTop + MenuHeight);
710         outtextxy(menuLeft + 15, MenuTop + 6, s[i]);
711         menuLeft = menuLeft + MenuWidth + MenuSpacing;
712     }
713 }
714
715 /*插入*/
716
717 info addinfo(info * newPeople)
718 {
719     bool ifaddName;
720     bool ifaddDataBirth;
721     bool ifaddPlaceBirth;
722     bool ifaddSex;
723     bool ifaddProfession;
724     bool ifaddSpouseName;
725     bool ifaddparent;
726     bool ifgetfather;
727
```

```
728     ifgetfather= InputBox(getfather, 20, "", "请输入你要添加的成员的父亲", "查询",
          300);
729     sscanf(getfather, "%s", findfather);   /*findfather 是根据用户输入的父亲姓名，用
          来作为findname()的参数找到要插入的人的父亲节点*/
730
731     /*输入要插入的节点信息*/
732     newPeople = (info*)malloc(sizeof(info));
733     if (ifgetfather)
734     {
735         ifaddName = InputBox(addName, 20, "", "请输入你要添加的成员的姓名", "查询",
              300);
736         sscanf(addName, "%s", newPeople->name);   /*获取用户输入的字符串*/
737     }
738     if (ifaddName)
739     {
740         ifaddDataBirth = InputBox(addData_birth, 20, "", "请输入出生日期", "查询",
              300);
741         sscanf(addData_birth, "%s", newPeople->data_birth);   /*获取用户输入的字符串
              */
742     }
743     if (ifaddDataBirth)
744     {
745         ifaddPlaceBirth= InputBox(addPlace_birth, 20, "", "请输入出生地", "查询",
              300);
746         sscanf(addPlace_birth, "%s", newPeople->place_birth);   /*获取用户输入的字符
              串*/
747     }
748     if (ifaddPlaceBirth)
749     {
750         ifaddSex= InputBox(addSex, 20, "", "请输入性别", "查询", 300);
751         sscanf(addSex, "%d", newPeople->sex);   /*获取用户输入的字符串*/
752     }
753     if (ifaddSex)
754     {
755         ifaddProfession= InputBox(addProfession, 20, "", "请输入职业", "查询", 300);
756         sscanf(addProfession, "%s", newPeople->profession);   /*获取用户输入的字符串
              */
757     }
758     if (ifaddProfession)
759     {
760         ifaddSpouseName = InputBox(addSpouse_name, 20, "", "请输入配偶", "查询",
              300);
761         sscanf(addSpouse_name, "%s", newPeople->spouse_name);   /*获取用户输入的字符
              串*/
762     }
763     if (ifaddSpouseName)
764     {
765         ifaddparent = InputBox(addParent_name, 20, "", "请输入父母姓名", "查询",
              300);
766         sscanf(addParent_name, "%s", newPeople->parent_name);   /*获取用户输入的字符
              串*/
767     }
768     if (ifaddparent)
769     {
770
771         LOGFONT f;
```

```
772         gettextstyle(&f);
773         f.lfHeight =35;
774         settextcolor(BLACK);
775         setbkmode(TRANSPARENT);
776         _tcscpy_s(f.lfFaceName, _T("黑体"));
777         settextstyle(&f);
778         char add[] = "新成员信息录入成功\n欢迎新成员加入！！";
779         RECT r = { Frame + 15, 75,width - 135,  height - 215 };
780         drawtext(add, &r, DT_CENTER | DT_VCENTER);
781     }
782     return *newPeople;
783 }
784
785 /*  绘制关系菜单*/
786 void relationshipMenu()
787 {
788     int g;
789     LOGFONT f;
790     gettextstyle(&f);                       // 获取当前字体设置
791     f.lfHeight = 25;
792     settextcolor(BLACK);
793     setbkmode(TRANSPARENT);
794     _tcscpy_s(f.lfFaceName, _T("黑体"));    // 设置字体为“黑体”
795     settextstyle(&f);
796     char r0[15] = "查找父母";
797     char r1[15] = "查找祖先";
798     char r2[15] = "查找兄弟";
799     char r3[15] = "查找孩子";
800     char r4[15] = "查找后代";
801     char r5[15] = "个人信息";
802     char *p[] = {r0,r1,r2 ,r3,r4,r5};
803     int relaMenuLeft = Frame + (width - 2 * Frame) / 3;   /*定义关系菜单左侧坐标*/
804     int relaMenuRight = relaMenuLeft + relationMenuWidth;   /*定义关系菜单右侧左边*/
805     int relaMenuTop = 63;   /*定义第一个关系按钮顶部坐标*/
806     int relaMenuButton = relaMenuTop + relationMenuHeight;   /*定义关系按钮底部坐标*/
807     fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清空输出区域*/
808     for ( g = 0; g < relationMenuNum; g++)
809     {
810         setfillcolor(RGB(72, 81, 81));
811         fillrectangle(relaMenuLeft, relaMenuTop, relaMenuRight, relaMenuButton);
812         outtextxy(relaMenuLeft+120, relaMenuTop + 26, p[g]);
813         //setfillcolor(RGB(255, 255, 255));
814         relaMenuTop = relaMenuTop + relationMenuHeight+3;
815         relaMenuButton = relaMenuTop + relationMenuHeight;
816     }
817     setfillcolor(RGB(255,255,255 ));
818 }
819
820
821
822 /* 鼠标事件*/
823
824 char * GetMouse(char out[500] )
825 {
826     person * t;
827     char file[20] = "02.txt";
```

```
828     FILE *fp = fopen(file, "r");
829     t = create_family(fp);
830     MOUSEMSG mousemsg;      /* 定义鼠标消息*/
831     while (true)
832     {
833         mousemsg = GetMouseMsg();   /*获取一条鼠标消息*/
834         int x, y;
835         bool mklButton;
836         x = mousemsg.x;   /*获取鼠标当前x坐标*/
837         y = mousemsg.y;   /*获取鼠标当前y坐标*/
838         mklButton = mousemsg.mkLButton;   /*获取鼠标当前左键是否按下*/
839
840
841         if ((y > height - 2 * Frame + Buttom) && (y < height - 2 * Frame + Buttom +
              buttleHeight))
842         {
843             /*查询*/
844             if ((x > Frame + Spacing / 2) && (x < Frame + Spacing / 2 + buttleWidth)
                  && mklButton)   /*鼠标在第一个按钮上单击左键时，激发鼠标事件 1*/
845             {
846                 fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清空输
                      出区域*/
847                 MouseFindNum++;
848                 if (MouseFindNum % 2 != 0)
849                 {
850                     queryMenu();
851                 }
852                 else   /*再次点击时重新绘制界面，关闭菜单*/
853                 {
854                     fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清
                          空输出区域*/
855                     initImg();
856                     DrawButtle();
857                     GetMouse(out);
858                 }
859             }
860             /*插入*/
861             else if ((x > (Frame + Spacing / 2) + buttleWidth + Spacing) && (x <
                  (Frame + Spacing / 2) + 2 * buttleWidth + Spacing) && mklButton)
862             {
863                 fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清空输
                      出区域*/
864                 info *newone= (info*)malloc(sizeof(info));
865                 newone = &addinfo(newpeople);   /*newonw就是要插入的节点信息*/
866                 FlushMouseMsgBuffer();   /*清空鼠标缓存区*/
867                 /****************************************
868                 *                                      *
869                 *          插入的代码                  *
870                 *          变量名：newone              *
871                 *                                      *
872                 ****************************************/
873             }
874             /*退出*/
875             else if ((x > (Frame + Spacing / 2) + 2 * (buttleWidth + Spacing)) && (x
                  < (Frame + Spacing / 2) + 2 * (buttleWidth + Spacing) + buttleWidth) &&
                   mklButton)
```

```
876             {
877                 exit(1);
878             }
879             /*画树*/
880             else if ((x > (Frame + Spacing / 2) + 3 * buttleWidth + Spacing) && (x < (Frame + Spacing / 2) + 3 * (buttleWidth + Spacing) + buttleWidth) && mklButton)
881             {
882                 LOGFONT f;
883                 gettextstyle(&f);
884                 f.lfHeight = 15;
885                 settextcolor(BLUE);
886                 setbkmode(TRANSPARENT);
887                 _tcscpy_s(f.lfFaceName, _T("黑体"));
888                 settextstyle(&f);
889                 setfillcolor(RGB(0,0,0));
890                 fillrectangle(0, 60, width, height - 2 * Frame);
891                 MouseTreeNum++;
892                 setlinecolor(BLUE);
893                 if (MouseTreeNum % 2 != 0)
894                 {
895                     imctree(t, 640, 60+10+5);
896                     setfillcolor(RGB(255, 255, 255));
897                 }
898                 else
899                 {
900                     setfillcolor(RGB(255, 255, 255));
901                     setlinecolor(RGB(255, 255, 255));
902                     fillrectangle(Frame, 60, width - 100, height - 2 * Frame);   /*清空输出区域*/
903                     initImg();
904                     DrawButtle();
905                     GetMouse(out);
906                 }
907             }
908         }
909         /* 如果单机次数为奇数，说明菜单栏弹出，判断鼠标单击位置*/
910         if (MouseFindNum % 2 != 0)
911         {
912             LOGFONT f;
913             gettextstyle(&f);                       // 获取当前字体设置
914             f.lfHeight = 25;
915             settextcolor(BLACK);
916             setbkmode(TRANSPARENT);
917             _tcscpy_s(f.lfFaceName, _T("黑体"));    // 设置字体为“黑体”
918             settextstyle(&f);
919 
920             if (y > (height - 200 + Buttom + buttleHeight + 20) && y < (height - 200 + Buttom + buttleHeight + 20 + MenuHeight))
921             {
922                 LOGFONT f;
923                 gettextstyle(&f);                       // 获取当前字体设置
924                 f.lfHeight = 25; // 设置字体高度
925                 settextcolor(BLACK);
926                 setbkmode(TRANSPARENT);
927                 _tcscpy_s(f.lfFaceName, _T("黑体"));    // 设置字体为“黑体”
```

```
928                 settextstyle(&f);
929                 /*根据姓名查找*/
930                 if (x > (Frame + Spacing / 2) && x < (Frame + Spacing / 2 +
                      MenuWidth) && mklButton)
931                 {
932                     ifgetName=InputBox(name, 20, "", "请输入你要查询的姓名", "查询",
                          300);   /*用于以对话框形式获取用户输入*/
933                     sscanf(name, "%s", getInfo);   /*获取用户输入的字符串*/
934
935                     RECT r = { Frame + 15, 75,width - 135,  height - 215 };
936                     FlushMouseMsgBuffer();   /*清空鼠标缓存区*/
937                 }
938                 /*根据出生年月查找*/
939                 else if (x > (Frame + Spacing / 2) + MenuWidth + MenuSpacing && x <
                      (Frame + Spacing / 2) + 2 * MenuWidth + MenuSpacing && mklButton)
940                 {
941                     InputBox(year, 20, "", "请输入你要查询的出生年月", "查询", 300,
                          0, false);   /*用于以对话框形式获取用户输入*/
942                     sscanf(year, "%s", getInfo);
943                     findbirth(t, year);
944                     FlushMouseMsgBuffer();   /*清空鼠标缓存区*/
945                 }
946                 /*根据逝世日期查找*/
947                 else if (x > (Frame + Spacing / 2) + 2 * MenuWidth + 2*MenuSpacing &&
                       x < (Frame + Spacing / 2) + 3 * MenuWidth +2* MenuSpacing &&
                      mklButton)
948                 {
949                     InputBox(finddeath, 20, "", "请输入你要查询的逝世年月", "查询",
                          300, 0, false);   /*用于以对话框形式获取用户输入*/
950                     sscanf(finddeath, "%s", getInfo);
951                     fdeathdata(t,finddeath);
952                     FlushMouseMsgBuffer();   /*清空鼠标缓存区*/
953                 }
954                 /*根据职业查找*/
955                 else if (x > (Frame + Spacing / 2) + 3*MenuWidth +3* MenuSpacing && x
                       < (Frame + Spacing / 2) + 4 * MenuWidth + 3 * MenuSpacing &&
                      mklButton)
956                 {
957                     InputBox(findprofe, 20, "", "请输入你要查询的职业", "查询", 300,
                          0, false);   /*用于以对话框形式获取用户输入*/
958                     sscanf(findprofe, "%s", getInfo);
959                     findpro(t, findprofe);
960                     FlushMouseMsgBuffer();   /*清空鼠标缓存区*/
961                 }
962                 /*根据配偶姓名查找*/
963                 else if (x > (Frame + Spacing / 2) + 4 * MenuWidth +4 * MenuSpacing
                      && x < (Frame + Spacing / 2) + 5 * MenuWidth + 4*MenuSpacing &&
                      mklButton)
964                 {
965                     InputBox(findspouse, 20, "", "请输入你要查询的逝世年月", "查询",
                          300, 0, false);   /*用于以对话框形式获取用户输入*/
966                     sscanf(findspouse, "%s", getInfo);
967                     findspous(t, findspouse);
968                     FlushMouseMsgBuffer();   /*清空鼠标缓存区*/
969                 }
970             }
```

```
971         }
972         if (ifgetName)   /* 如果输入要查询的姓名后单击确定，则输出关系菜单供选择*/
973         {
974
975             relationshipMenu();
976             ifgetName = FALSE;   /*初始化，避免重复查找不输入*/
977             Sleep(60);
978             MOUSEMSG mousemsg1;     /* 定义鼠标消息*/
979             person *str;
980             str = findname(t, name);
981             while (true)
982             {
983                 mousemsg1 = GetMouseMsg();   /*获取一条鼠标消息*/
984                 int x1, y1;
985                 bool mklButton1;
986                 bool ifgetGener;
987                 x1 = mousemsg1.x;   /*获取鼠标当前x坐标*/
988                 y1 = mousemsg1.y;   /*获取鼠标当前y坐标*/
989                 mklButton1 = mousemsg1.mkLButton;   /*获取鼠标当前左键是否按下*/
990
991                 if (x1 > Frame + (width - 2 * Frame) / 3 && x1 < Frame + width - 2 *
                      Frame / 3 + relationMenuWidth)   /*判断鼠标是否在关系菜单的宽度区间
                      */
992                 {
993                     if ((y1 > 63) && (y1 < (63 + relationMenuHeight)) &&
                          mklButton1)   /*单击第一个按钮查询父母*/
994                     {
995                         fillrectangle(Frame, 60, width - 100, height - 2 *
                          Frame);   /*清空输出区域*/
996                         parents(t, str);
997                         break;
998                     }
999                     else if ((y1 > (63 + relationMenuHeight + 3)) && (y1 < (63 + 2 *
                      relationMenuHeight + 3)) && mklButton1)   /*单击第二个按钮，查询
                      祖先*/
1000                    {
1001                        fillrectangle(Frame, 60, width - 100, height - 2 *
                          Frame);   /*清空输出区域*/
1002                        ancestor(t, str);
1003                        break;
1004                    }
1005                    else if ((y1 > (63 + 2 * relationMenuHeight + 6)) && (y1 < (63 +
                      3 * relationMenuHeight + 6)) && mklButton1)   /*单击第三个按钮，
                      查询兄弟*/
1006                    {
1007                        fillrectangle(Frame, 60, width - 100, height - 2 *
                          Frame);   /*清空输出区域*/
1008                        brother(t, str);
1009                        break;
1010                    }
1011                    else if ((y1 > (63 + 3 * relationMenuHeight + 9)) && (y1 < (63 +
                      4 * relationMenuHeight + 9)) && mklButton1)   /*单击第四个按钮，
                      查询孩子*/
1012                    {
1013                        fillrectangle(Frame, 60, width - 100, height - 2 *
                          Frame);   /*清空输出区域*/
```

```
1014                        children1(str);
1015                        break;
1016                    }
1017                    else if ((y1 > (63 + 4 * relationMenuHeight + 12)) && (y1 < (63 +
                          5 * relationMenuHeight + 12)) && mklButton1)   /*单击第五个按
                        钮，查询后代*/
1018                    {
1019                        fillrectangle(Frame, 60, width - 100, height - 2 *
                        Frame);   /*清空输出区域*/
1020                        ifgetGener = InputBox(getGeneration, 20, "", "您要查询第几
                        代", "查询", 300);
1021                        sscanf(getGeneration, "%c",gener);
1022                        if (ifgetGener)
1023                        {
1024                            person *find_name = findname(t, name);
1025                            children(find_name, gener);
1026                        }
1027                        break;
1028                    }
1029                    else if ((y1 > (63 + 5 * relationMenuHeight + 12)) && (y1 < (63 +
                          6 * relationMenuHeight + 12)) && mklButton1)   /*单击第6个按
                        钮，查看个人信息*/
1030                    {
1031                        fillrectangle(Frame, 60, width - 100, height - 2 *
                        Frame);   /*清空输出区域*/
1032
1033                        display(str);
1034                        break;
1035                    }
1036                }
1037            }
1038        }
1039    }
1040    return getInfo;
1041 }
1042
1043
1044 void title()
1045 {
1046     LOGFONT f;
1047     gettextstyle(&f);                       // 获取当前字体设置
1048     f.lfHeight = 35; // 设置字体高度为 12
1049     settextcolor(RGB(72,81,81));
1050     setbkmode(TRANSPARENT);
1051     _tcscpy_s(f.lfFaceName, _T("黑体"));    // 设置字体为“黑体”
1052     settextstyle(&f);
1053     char s[] = "家庭族谱查询";
1054     outtextxy(15,15, s);
1055 }
1056
1057 int main()
1058 {
1059     char * info[100];
1060     char out[500];   /*之后把输出结果赋值给out数组*/
1061     initgraph(width, height);   /*创建绘图窗口*/
1062     initImg();
```

```
1063        title();
1064        DrawButtle();
1065        info[100]=GetMouse(out);    /*info数组为输入内容*/
1066        _getch();                  // 按任意键继续
1067        closegraph();               // 关闭绘图窗口
1068    }
1069
```