



Sudoku

实验报告

1551265 计 1 张伯阳



1. Sudoku (数独)

1.1. 操作方法及要求

【游戏规则:】

数独是源自18世纪瑞士的一种数学游戏。是一种运用纸、笔进行演算的逻辑游戏,我们现将其移植到计算机上。

玩家需要根据9×9盘面上的已知数字,推理出所有剩余空格的数字,并满足每一行、每一列、每一个粗线宫(3×3)内的数字均含1-9,不重复。

数独盘面是个九宫,每一宫又分为九个小格。在这八十一格中给出一定的已知数字和解题条件,利用逻辑和推理,在其他的空格上填入1-9的数字。使1-9每个数字在每一行、每一列和每一宫中都只出现一次,所以又称“九宫格”。

【显示要求:】

1. 字符界面下的数独求解

- 输入数独题目文件名,从文件中读入初始数据,要判断文件内容是否合法(是否都是1-9)从文件中读入后,要判断初始是否有冲突,如果出错,要给出相应的错误提示
- 用不同颜色表示已有的固定数字、未填的0以及填写的数字,如果填写的数字导致行/列/九宫格有冲突,则用不同颜色标识
- 用指定格式输入解(例如:a76表示a行7列输入6),判断输入的正确性,填写到指定位置
- 要记录每一步操作(正确/错误),并可以依次回退
- 成功则给出提示信息

2. cmd伪图形界面下的数独求解

- 获取当前目录下所有sudoku开头的文件名,在cmd下模拟一个列表框,将文件排列在列表框中,用上下箭头键移动,回车键选择所需要的文件,从中读取题目信息(如果文件名数量超过列表框的行数,则需要上下滚动)
- 其余要求同子题目1(不同颜色标注、输入解、支持回退、成功给出提示)

3. 在图形界面下自动求解

- 自动解可以采用递归算法,也可以自己定义搜索算法

2. 整体设计思路

按照实验的功能顺序分步骤设计, 较分立的功能设单独的函数进行操作, 本着尽量共用函数的准则进行接下来的函数设计. 第1个功能不需要图形化操作, 优先设计. 定义出储存地图的数组 `a[11][11]`, 冲突错误数组 `err[11][11]` 表示具体冲突的格子, 冲突行 `err_x[11]`, 冲突列 `err_y[11]`, 冲突九宫格 `err_squa[9]`. `step` 结构体用来储存历史记录, 方便撤回, 撤回时因为不需要重做所以马上删除撤回后一步防止内存浪费, 储存记录时按需申请. **a数组开设11行11列和 `err_x`, `err_y` 开设11的目的是为了将边缘行列初始化为-1, 便于判断边界并且可以有效地防止越界. 浪费空间不多却效果显著, 为了区分可修改数据和初始数据, 在代码设计时将初始非零值即不可改变的值在地图上置原值, 后来修改的值和初始的零值即可修改值置10+该实际值加以区分, 虽然这种方法可能不是最方便的方法, 不过可以有效防止对不可修改的值进行修改, 在取值时也只需要将a数组值%10即可.** 因为还需要判断和打印冲突行列九宫格不同颜色, 冲突格子不同颜色, 所以需要开设 `err` 数组进行储存. 大部分复杂功能的实现在功能1已经设计完成, 所以功能2只需要打印出图形界面即可, 主要的工作量都体现在打印上下翻的文件表上. 功能3使用暴力尝试的方法就将难度降低了很多, 只需要写一个 `auto_solve` 函数而且不需要太多的代码量.

3. 主要功能的实现

- 1功能直接构造相应的地图数组并判断是否有冲突, 若有冲突, 修改相应的 `err` 数组, 并打印
- 2功能只需要打印出相应的图形界面, 除了选择文件表以外 其他和上一功能基本无差
- 3功能主要是构造一个 `auto_solve` 函数进行递归尝试, 并将可能的地图打印出来

4. 调试过程碰到的问题

碰到的问题主要有下:

1. 冲突数据的处理

在经过一番考虑和实验后决定使用4个 `err` 系列数组分别储存冲突点冲突行列冲突九宫格, 使得所有输出的要求都能够得到满足.

2. 撤回的处理

处理撤回 (`bk`) 操作的本质就是进行历史数据的有效储存, 如何进行每一步的按顺序储存并能够成功显示之前的操作点 `xy` 和具体值并且不浪费内存. 我最终决定使用逆序链表, 即链表数据域储存 `xy` 和上一步的值, 指针域储存上一个结点指针, 因为我只需要撤回上一步的操作而不需要重做, 所以可以撤回后马上删除后一结点, 用链表操作的最大好处就是编写和操作简便, 只需要判断当前结点是否为头结点即可判断是否可以撤回.

3. 数组越界

这次代码完成用了大半天, `debug` 用了一天多, 主要的 `debug` 时间就都花在这里, 3功能每次自动操作到最后报数组越界弹窗, 因为使用递归函数, 进行的步骤过多, 很难一步一步调试寻找 `bug`, 所以 `debug` 的过程变得十分的漫长. 最后是在同学的提示将 `err_squa` 数组增大到12就不再弹窗, 而我之前试验过将原来9的数组扩大到11却没有作用, 这个12马上让我意识到哪里出现了问题. 在改动之后的 `err_squa` 的计算九宫格序号式子后, 函数最前面清零处忘记修改. 但是由于清零的这个问题不会影响到整个函数的结果即不会影响到自动填充数独程序的执行, 所以填充能够成功完成, 但是到了 `auto_solve` 返回时, 由于之前的 `err_squa` 数组非法占用了内

存, 弹窗报错.

5. 心得体会

从这次作业我得到的主要教训有下:

1. 写之前多考虑一些之后的可能情况

在写后几个图形界面函数时很明显的遇到了很多次的问题就是之前在写这个函数时只想到了实现眼下的功能所需要的参数进行传入, 后续功能或者完善程序时发现还需要别的参数, 于是又要修改, 这些函数很多还是中间函数, 修改传参就要从前面一个一个函数依次修改, 浪费了一些时间.

2. 多加注释 多使用宏定义和常变量

加注释的作用很明显: 让阅读的人更容易看懂, 包括自己之后debug时也会变得容易.

为了达到这个目的, 定义函数名参数名也应该避免使用fun, a, b, c这样的名字, 在一个大程序中这样的参数多了会让阅读变得非常困难. 使用宏定义常变量的目的, 一是为了让阅读变得方便, 一个OK, ERROR肯定要比1, 0看起来更加易懂, -1, -2等特殊返回值也变得更直观. 二是为了让修改方便, 一个使用很多次的常量如果要修改只需要在.h里面修改一次即可实现, 使用宏定义常变量也可以让计算式变得易写易懂不易出错. 因为这方面的练习较少, 我在这次作业中的使用还很不好, 图形界面的计算还是在用常数实现, 以后需要多加注意.

3. 保证一个程序的统一

暂且不说养成平时的习惯, 我这次在写这一个程序中也没有做到前后统一. 有些地方使用x, y顺序, 有些地方使用y, x顺序使传参的过程思考量变大, 绕来绕去容易出错.

4. 修改时瞻前顾后

修改时看看函数的前后是否有需要牵连修改的地方, 而不是只有编译报错才会注意到去修改.

5. 提高debug效率

Bug的出现是难免的, 但是如何快速的找出bug并改正算是比写代码还要更高难度的技能.

积累错误修改经验, 巧妙设置断点和通过弹窗报错行按需增加数组的输出或者对可能出错的数组越界问题进行检查都是将来debug过程中需要注意的.

这次的大作业的完成, 我用了两天时间, 但是纯写代码还是只用了不到一天时间, 剩下的时间几乎都是在debug, 说明我的代码在完成的过程中没有考虑到可能出现的bug, 和如果出现bug的处理方法, 毕竟bug几乎是难以避免的, 但是如何减少bug, 有意把可能出现bug的地方暴露出来和在出现了bug后将debug时间减少到最短 比写代码还要重要. 这是以后要注意的, 并不是写代码写得快一些就可以完事.

6. 附件：源程序

```

/*1551265 计1 张伯阳*/
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <iomanip>
#include <ctime>
#include <cstdio>
#include <stdlib.h>
#include <conio.h>
#include <Windows.h>
#include <fstream>
#include <io.h>
#include "cmd_console_tools.h"
using namespace std;

#define length 100
#define hight 40
#define pertime 5
#define XX 9
#define YY 9

struct step {
    int x, y, data;
    struct step *pre;
};

const char file_name[15] = "sudoku*.txt";
const HANDLE hout = GetStdHandle(STD_OUTPUT_HANDLE); //取标准输出设备对应的句柄
const HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); //取标准输入设备对应的句柄
void readl_1(int *choice);
int read_data(int a[][11], char file[]);
void read_name(char name[][20], char file[]);
bool read_name_(char name[][20], char file[]); int filesearch(char path[], char mode[], char
name[][20]);
int read_pre(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[]); int
console_file(char file[][20], int line);
void choicel(int a[][11]);
void choice2(int a[][11]);
void choice3(int a[][11]);
int judge_start(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[]);
int judge_in(int a[][11], int x, int y, int num, bool err[][11], bool err_x[], bool err_y[],
bool err_squa[]);
bool judge_finish(int a[][11]);
void showch_(const HANDLE hout, const char ch, const int bg_color = COLOR_BLACK, const int
fg_color = COLOR_WHITE);
void printarr(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[]);
void print_pre(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[]);
int play(step **p, step *head, step **q, int a[][11], bool err[][11], bool err_x[], bool
err_y[], bool err_squa[], int num);
void clr_a(int a[]);
int auto_solve(int a[][11], int x, int y, int *n);

```

```

/*1551265 计1 张伯阳*/
#include "90-b2.h"
int main()
{
    int a[XX + 2][YY + 2], choice;
    const HANDLE hout = GetStdHandle(STD_OUTPUT_HANDLE); //取标准输出设备对应的句柄
    const HANDLE hin = GetStdHandle(STD_INPUT_HANDLE); //取标准输入设备对应的句柄
    setcolor(hout, 0, 7);
    srand((unsigned)time(NULL));
    while (1)
    {
        memset(a, -1, sizeof(a));
        setconsoleborder(hout, length, height, height);
        readl_1(&choice);
        if (!choice)
            break;
        system("cls");
        if (choice == 1)
            choice1(a);
        if (choice == 2)
            choice2(a);
        if (choice == 3)
            choice3(a);
        setcolor(hout, 0, 7);
        if (choice != 1)
            gotoxy(hout, 1, 24);
        cout << "本小题结束, 请输入回车键继续..." << endl;
        while (_getch() != '\r');
        setfontsize(hout, L"新宋体", 18);
        system("cls");
    }
}

void readl_1(int *choice)
{
    while (1)
    {
        cout << "-----" << endl;
        cout << "1. 字符界面游戏(带回退功能)\n2. 图形界面游戏(带回退功能)\n3. 图形界面自动解\n(显示过程并带延时)\n0. 退出\n";
        cout << "-----" << endl;
        char ch = _getch();
        if (ch >= '0' && ch <= '3')
        {
            *choice = int(ch - '0');
            break;
        }
    }
}

/*1551265 计1 张伯阳*/
#include "90-b2.h"

int console_file(char file[][20], int line)

```

```
{
    setcursor(hout, CURSOR_INVISIBLE);
    int i, j = 0, k;
    const int l = 2 + 4 * 9;
    const int r = 2 + 2 * 9;
    gotoxy(hout, l + 6, 2);
    cout << "数独文本文件";
    gotoxy(hout, l + 4, 3);
    cout << "┌───────────┐";
    for (i = 0; i < 5; i++)
        if (i != line - 1 && line)
        {
            gotoxy(hout, l + 4, 4 + i);
            cout << "│" << setw(20) << setiosflags(ios::left) << file[i] << "│";
        }
        else
            break;
    gotoxy(hout, l + 4, 4 + i);
    cout << "└───────────┘";
    if (line == 0)
        return -1;
    gotoxy(hout, l + 6, 4);
    setcolor(hout, 7, 0);
    cout << setw(20) << setiosflags(ios::left) << file[0];
    while (1)
    {
        int a, b;
        a = _getch();
        if (a == 13)
            return j;
        else if (a == 224)
            b = _getch();
        while (a != 224 || (b != 72 && b != 75 && b != 77 && b != 80))
        {
            cin.clear();
            cin.sync();
            a = _getch();
            if (a == 13)
                return j;
            else if (a == 224)
                b = _getch();
        }
        if (b == 72)
        {
            if (j > 0)
            {
                if (j > i - 5)
                {
                    if (i <= 5)
                        gotoxy(hout, l + 6, 4 + j);
                    else
                        gotoxy(hout, l + 6, 9 + j - i);
                    setcolor(hout, 0, 7);
                    cout << setw(20) << setiosflags(ios::left) << file[j];
                }
            }
        }
    }
}
```

```

        j--;
        if (i < 5)
            gotoxy(hout, 1 + 6, 4 + j);
        else
            gotoxy(hout, 1 + 6, 9 + j - i);
        setcolor(hout, 7, 0);
        cout << setw(20) << setiosflags(ios::left) << file[j];
    }
    else if (j == i - 5)
    {
        j--, i--;
        for (k = j; k < j + 5; k++)
            if (strcmp(file[k], "NULL"))
            {
                gotoxy(hout, 1 + 6, 4 + k - j);
                setcolor(hout, 0, 7);
                cout << setw(20) << setiosflags(ios::left) << file[k];
            }
            else
                break;
        gotoxy(hout, 1 + 6, 4);
        setcolor(hout, 7, 0);
        cout << setw(20) << setiosflags(ios::left) << file[j];
    }
}
else if (b == 80)
{
    if (j < line - 2)
    {
        if (j < i - 1)
        {
            if (i <= 5)
                gotoxy(hout, 1 + 6, 4 + j);
            else
                gotoxy(hout, 1 + 6, 9 + j - i);
            setcolor(hout, 0, 7);
            cout << setw(20) << setiosflags(ios::left) << file[j];
            j++;
            if (i <= 5)
                gotoxy(hout, 1 + 6, 4 + j);
            else
                gotoxy(hout, 1 + 6, 9 + j - i);
            setcolor(hout, 7, 0);
            cout << setw(20) << setiosflags(ios::left) << file[j];
        }
        else if (j == i - 1)
        {
            j++;
            i++;
            for (k = 0; k < 5; k++)
                if (strcmp(file[j - 4 + k], "NULL"))
                {
                    gotoxy(hout, 1 + 6, 4 + k);

```



```

        setcolor(hout, 0, 7);
        cout << setw(20) << setiosflags(ios::left) << file[j - 4 +
k];
    }
    else
        break;
    gotoxy(hout, 1 + 6, 8);
    setcolor(hout, 7, 0);
    cout << setw(20) << setiosflags(ios::left) << file[j];
}
}
}
return 0;
}

void showch_(const HANDLE hout, const char ch, const int bg_color, const int fg_color)
{
    setcolor(hout, bg_color, fg_color);
    cout << ch;
}

void print_pre(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[])
{
    int i, j, l = 2 + 4 * 9, r = 2 + 2 * 9;
    gotoxy(hout, 4, 3);
    setcolor(hout, 15, 0);
    for (i = 1; i < r; i++)
    {
        for (j = 0; j < l; j += 2)
        {
            gotoxy(hout, j + 4, i + 2);
            if (i == 1 && j == 0)
                cout << "┐";
            else if (i == 1 && j == l - 2)
                cout << "┌";
            else if (i == r - 1 && j == 0)
                cout << "└";
            else if (i == r - 1 && j == l - 2)
                cout << "┘";
            else if (j == 0 || j == l - 2)
            {
                if (i % 2 == 1 && j == 0)
                {
                    if (i % 6 == 1)
                        cout << "┐";
                    else
                        cout << "┌";
                }
                else if (i % 2 == 1 && j == l - 2)
                {
                    if (i % 6 == 1)
                        cout << "└";
                    else

```

```

        cout << "┘";
    }
    else
        cout << "┐";
}
else if (i == 1 || i == r - 1)
{
    if (j % 4 == 0 && i == 1)
    {
        if (j % 12 == 0)
            cout << "┘";
        else
            cout << "┐";
    }
    else if (j % 4 == 0 && i == r - 1)
    {
        if (j % 12 == 0)
            cout << "┘";
        else
            cout << "┐";
    }
    else
        cout << "═";
}
else if (j % 4 == 0 || i % 2 == 1)
{
    if (j % 4 == 0 && i % 2 == 1)
    {
        if (j % 12 == 0 && i % 6 == 1)
            cout << "┘";
        else if (j % 12 == 0 && i % 6 != 1)
            cout << "┘";
        else if (j % 12 && i % 6 == 1)
            cout << "┘";
        else
            cout << "┘";
    }
    else if (j % 4 == 0 && i % 2 != 1)
    {
        if (j % 12 == 0)
            cout << "┐";
        else
            cout << "┐";
    }
    else
    {
        if (i % 6 == 1)
            cout << "═";
        else
            cout << "═";
    }
}
else
    cout << " ";

```

```

    }
}
for (i = 1; i <= 9; i++)
    for (j = 1; j <= 9; j++)
        if (!(a[j][i] % 10) && (err_x[i] || err_y[j] || err_squa[((i - 1) / 3) * 3 + (j
- 1) / 3]))
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 14, 1);
        else if (!(a[j][i] % 10))
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 15, 1);
        else if (a[j][i] < 10 && (err_x[i] || err_y[j] || err_squa[((i - 1) / 3) * 3 + (j
- 1) / 3]))
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 14, 8);
        else if (a[j][i] < 10)
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 15, 8);
        else if (!err[j][i] && (err_x[i] || err_y[j] || err_squa[((i - 1) / 3) * 3 + (j
- 1) / 3]))
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 14, 2);
        else if (err[j][i])
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 14, 12);
        else
            showch(hout, 4 * j + 2, 2 * i + 2, a[j][i] % 10 + '0', 15, 2);
setcolor(hout, 0, 7);
for (int i = 1; i <= 9; i++)
{
    gotoxy(hout, 2 + 4 * i, 2);
    cout << char('a' + i - 1);
    gotoxy(hout, 2, 2 + 2 * i);
    cout << i;
}

}

void choice2(int a[][11])
{
    bool err[11][11], err_x[11] = { 0 }, err_y[11] = { 0 }, err_squa[9] = { 0 }, ok = 0;
    memset(err, 0, sizeof(err));
    step *p, *head, *q;
    if (read_pre(a, err, err_x, err_y, err_squa))
        return;
    head = p = q = new step;
    if (head == NULL)
        return;
    int num = 0;
    while (1)
    {
        print_pre(a, err, err_x, err_y, err_squa);
        gotoxy(hout, 0, 2 * 9 + 5);
        setcolor(hout, 0, 7);
        cout << "
" << endl << "
";
        gotoxy(hout, 0, 2 * 9 + 5);
        setcolor(hout, 0, 7);
        if (play(&p, head, &q, a, err, err_x, err_y, err_squa, num))
            break;
        Sleep(pertime * 300);
    }
}

```

```

    }
}

void choice3(int a[][11])
{
    bool err[11][11], err_x[11] = { 0 }, err_y[11] = { 0 }, err_squa[11] = { 0 };
    memset(err, 0, sizeof(err));
    step *p, *head;
    if (read_pre(a, err, err_x, err_y, err_squa))
        return;
    head = p = new step;
    if (head == NULL)
        return;
    int num = 0, n = 0;
    print_pre(a, err, err_x, err_y, err_squa);
    while (_getch() != '\r');
    if (judge_finish(a))
    {
        cout << "完整数独" << endl;
        return;
    }
    int temp_x = 0, temp_y = 0;
    for (int j = 1; j <= 9; j++)
    {
        for (int k = 1; k <= 9; k++)
            if (!(a[k][j] % 10))
            {
                temp_x = k, temp_y = j;
                break;
            }
        if (temp_x)
            break;
    }
    setcursor(hout, CURSOR_INVISIBLE);
    if (!auto_solve(a, temp_x, temp_y, &n))
    {
        setcolor(hout, 0, 7);
        gotoxy(hout, 15, 23);
        cout << "无解" << endl;
    }
    else
    {
        print_pre(a, err, err_x, err_y, err_squa);
        gotoxy(hout, 15, 23);
        cout << "数独完成" << endl;
    }
    setcursor(hout, CURSOR_VISIBLE_NORMAL);
}

/*1551265 计1 张伯阳*/
#include "90-b2.h"
void printarr(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[])
{
    int i, j;

```

```

char ch;
cout << " | ";
for (i = 1; i <= XX; i++)
    cout << ' ' << i << ' ';
cout << endl;
cout << "----";
for (i = 1; i <= XX; i++)
    cout << "----";
cout << endl;
for (i = 1; i <= YY; i++)
{
    ch = 'a' + i - 1;
    cout << ch << " | ";
    for (j = 1; j <= XX; j++)
    {
        cout << ' ';
        if (a[j][i] % 10 == 0)
            showch(hout, a[j][i] % 10 + '0', (err_x[i] || err_y[j] || err_squa[(i -
1) / 3 * 3 + (j - 1) / 3]) * 15, 6);
        else
            showch(hout, a[j][i] % 10 + '0', (err_x[i] || err_y[j] || err_squa[(i -
1) / 3 * 3 + (j - 1) / 3]) * 15, 9 + err[j][i] * 3);
        setcolor(hout, 0, 7);
        cout << ' ';
    }
    cout << endl;
}
cout << endl;
}

void choice1(int a[][11])
{
    char file[20];
    do {
        char name[15][20];
        read_name(name, file);
    } while (read_data(a, file));
    cout << "读入的数据为: " << endl;
    int num = 0;
    step *p, *head, *q;
    bool err[11][11], err_x[11] = { 0 }, err_y[11] = { 0 }, err_squa[9] = { 0 }, ok = 0;
    memset(err, 0, sizeof(err));
    if (num = judge_start(a, err, err_x, err_y, err_squa))
        cout << "检查后的数据有" << num << "处冲突, 具体是: " << endl;
    if (num)
    {
        printarr(a, err, err_x, err_y, err_squa);
        cout << "原始数据文件有错, 请修改正确后继续" << endl;
        return;
    }
    if (judge_finish(a))
    {
        cout << "完整数独" << endl;
        return;
    }
}

```

装

订

线

```

    }
    head = p = q = new step;
    if (head == NULL)
        return;
    while (1)
    {
        printarr(a, err, err_x, err_y, err_squa);
        if (play(&p, head, &q, a, err, err_x, err_y, err_squa, num))
            break;
    }
}

/*1551265 计1 张伯阳*/
#include "90-b2.h"
void clr_a(int a[])
{
    for (int i = 0; i <= 9; i++)
        a[i] = 0;
}

int filesearch(char path[], char mode[], char name[][20])
{
    struct _finddata_t filefind;
    char curr[40];
    strcpy(curr, path);
    strcat(curr, "\\");
    strcat(curr, mode);
    int done = 0, handle, i = 0;
    handle = _findfirst(curr, &filefind);
    if (-1 == handle)
        return -1;
    do {
        if (!strcmp(filefind.name, ".."))
            continue;
        strcpy(curr, filefind.name);
        strcpy(name[i], curr);
        if (_A_SUBDIR == filefind.attrib)
            filesearch(curr, mode, name);
        i++;
    } while (!(done = _findnext(handle, &filefind)));
    _findclose(handle);
    return i;
}

int judge_start(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[])//
初始判断
{
    int flag[11], num = 0;
    //行判断
    for (int i = 1; i <= YY; i++)
    {
        clr_a(flag);
        for (int j = 1; j <= XX; j++)

```

```

        if (flag[a[j][i] % 10])
            num++, err_x[i] = 1, err[j][i] = 1;
        else if (a[j][i] % 10)
            flag[a[j][i] % 10]++;
    }
    //列判断
    for (int i = 1; i <= XX; i++)
    {
        clr_a(flag);
        for (int j = 1; j <= YY; j++)
            if (flag[a[i][j] % 10])
                num++, err_y[i] = 1, err[i][j] = 1;
            else if (a[i][j] % 10)
                flag[a[i][j] % 10]++;
    }
    //九宫格判断
    for (int i = 0; i < 9; i++)
    {
        clr_a(flag);
        for (int j = 1; j <= 3; j++)
            for (int k = 1; k <= 3; k++)
                if (flag[a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10])
                    num++, err_squa[i] = 1, err[(i % 3) * 3 + k][(i / 3) * 3 + j] = 1;
                else if (a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10)
                    flag[a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10]++;
    }
    return num;
}

int judge_in(int a[][11], int x, int y, int num, bool err[][11], bool err_x[], bool err_y[],
bool err_squa[])//输入判断
{
    int flag[11] = { 0 };
    err_x[y] = 0, err_y[x] = 0, err_squa[((y - 1) / 3) * 3 + (x - 1) / 3] = 0;
    for (int i = 1; i <= YY; i++)
        err[x][i] = 0, err[i][y] = 0;
    for (int j = 1, i = ((y - 1) / 3) * 3 + (x - 1) / 3; j <= 3; j++)
        for (int k = 1; k <= 3; k++)
            err[(i % 3) * 3 + k][(i / 3) * 3 + j] = 0;

    //行判断
    for (int j = 1; j <= XX; j++)
        if (flag[a[j][y] % 10])
        {
            num++, err_x[y] = 1, err[j][y] = 1;
            for (int k = 1; k <= XX; k++)
                if (a[j][y] % 10 == a[k][y] % 10)
                    err[k][y] = 1;
        }
        else if (a[j][y] % 10)
            flag[a[j][y] % 10]++;
    //列判断
    clr_a(flag);
    for (int j = 1; j <= YY; j++)
        if (flag[a[x][j] % 10])

```

```

{
    num++, err_y[x] = 1, err[x][j] = 1;
    for (int k = 1; k <= YY; k++)
        if (a[x][j] % 10 == a[x][k] % 10)
            err[x][k] = 1;
}
else if (a[x][j] % 10)
    flag[a[x][j] % 10]++;
//九宫格判断
clr_a(flag);
for (int j = 1, i = ((y - 1) / 3) * 3 + (x - 1) / 3; j <= 3; j++)
    for (int k = 1; k <= 3; k++)
        if (flag[a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10])
        {
            num++, err_squa[i] = 1, err[(i % 3) * 3 + k][(i / 3) * 3 + j] = 1;
            for (int jj = 1; jj <= 3; jj++)
                for (int kk = 1; kk <= 3; kk++)
                    if (a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10 == a[(i % 3) * 3 +
kk][(i / 3) * 3 + jj] % 10)
                        err[(i % 3) * 3 + kk][(i / 3) * 3 + jj] = 1;
        }
        else if (a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10)
            flag[a[(i % 3) * 3 + k][(i / 3) * 3 + j] % 10]++;
return num;
}

bool judge_finish(int a[][11])
{
    for (int i = 1; i <= 9; i++)
        for (int j = 1; j <= 9; j++)
            if (!(a[j][i] % 10))
                return 0;
    return 1;
}

int play(step **p, step *head, step **q, int a[][11], bool err[][11], bool err_x[], bool
err_y[], bool err_squa[], int num)
{
    cout << "请按行/列/值的方式输入(例如:5c6=第5行c列为6), 输入bk表示回退一次: "; //y-x-值
    char str[10];
    bool ok = 0;
    cin >> str;
    if (!strcmp(str, "bk"))
    {
        if (*p == head) //步数为0不能撤回
        {
            cout << "操作步骤已为空" << endl;
            return 0;
        }
        else
        {
            *q = *p;
            *p = (*p)->pre;
            delete[] * q;

```



```

        a[(*p)->x][( *p)->y] = (*p)->data + 10;
        num = judge_in(a, (*p)->x, (*p)->y, num, err, err_x, err_y, err_squa);
    }

}
else
{
    if (str[0] > '9' || str[0] < '1' || str[1] > 'i' || str[1] < 'a' || str[2] > '9' || str[2]
    < '0')
    {
        cout << "输入错误" << endl;
        return 0;
    }
    else
    {
        if (a[str[1] - 'a' + 1][str[0] - '0'] >= 10)
        {
            (*p)->x = str[1] - 'a' + 1, (*p)->y = str[0] - '0', (*p)->data = a[str[1]
            - 'a' + 1][str[0] - '0'] % 10;
            (*q) = (*p);
            (*p) = new step; //申请(*p)储存历史操作, (*p)百位表示x, 十位y值, 个位为之前值
            if ((*p) == NULL)
                return -1;
            (*p)->pre = *q;
            a[str[1] - 'a' + 1][str[0] - '0'] = str[2] - '0' + 10; //之后修改的值+10和
            原有值进行区分
            num = judge_in(a, str[1] - 'a' + 1, str[0] - '0', num, err, err_x, err_y,
            err_squa);
            if (!num) //判断完成
            {
                if (judge_finish(a))
                {
                    cout << "数独已完成" << endl;
                    return 1;
                }
            }
            else
                cout << "检测出数据有" << num << "处冲突" << endl;
        }
        else
            cout << "输入非法" << endl;
    }
}
return 0;
}

int auto_solve(int a[][11], int x, int y, int *n)
{
    if (x > 9 || x < 1 || y > 9 || y < 1)
        return 0;
    for (int i = 1; i <= 9; i++)
    {
        bool err[11][11], err_x[11] = { 0 }, err_y[11] = { 0 }, err_squa[11] = { 0 };
        memset(err, 0, sizeof(err));
    }
}

```

```

a[x][y] = i + 10;
if (!judge_in(a, x, y, 0, err, err_x, err_y, err_squa))
{
    (*n)++;
    showch(hout, 4 * x + 2, 2 * y + 2, a[x][y] % 10 + '0', 15, 4);
    gotoxy(hout, 1, 23);
    setcolor(hout, 0, 7);
    cout << "搜索" << *n << "次";
    Sleep(pertime);
    int temp_x = 0, temp_y = 0;
    if (judge_finish(a))
        return 1;
    for (int j = y; j <= 9; j++)
    {
        for (int k = 1; k <= 9; k++)
            if (!(a[k][j] % 10))
            {
                temp_x = k, temp_y = j;
                break;
            }
        if (temp_x)
            break;
    }
    if (auto_solve(a, temp_x, temp_y, n))
        return 1;
}
if (i == 9)
{
    a[x][y] = 10;
    showch(hout, 4 * x + 2, 2 * y + 2, a[x][y] % 10 + '0', 15, 2);
    Sleep(pertime);
}
}
return 0;
}

int read_pre(int a[][11], bool err[][11], bool err_x[], bool err_y[], bool err_squa[])
{
    char name[20][20], file[20];
    int num = 0;
    bool ok = 0;
    if (read_name_(name, file))
    {
        gotoxy(hout, 2, 2 * 9 + 5);
        cout << "未读取到sudoku为前缀的文件" << endl;
        return -1;
    }
    read_data(a, file);
    setcolor(hout, 0, 15);
    print_pre(a, err, err_x, err_y, err_squa);
    gotoxy(hout, 0, 2 * 9 + 5);
    setcolor(hout, 0, 7);
    if (num = judge_start(a, err, err_x, err_y, err_squa))
        cout << "检查后的数据有" << num << "处冲突" << endl;
}

```

```

    if (num)
    {
        cout << "原始文件有错, 修改正确后继续" << endl;
        return -1;
    }
    for (int i = 1, ok = 1; i <= XX; i++)
        for (int j = 1; j <= YY; j++)
            if (!a[i][j])
                ok = 0;
    if (ok)
    {
        cout << "完整数独" << endl;
        return 1;
    }
    return 0;
}

int read_data(int a[][11], char file[])
{
    fstream data;
    data.open(file, ios::in);
    if (!data.is_open())
    {
        cout << "文件无法打开" << endl;
        return -1;
    }
    for (int j = 1, temp; j <= YY; j++)
        for (int i = 1; i <= XX; i++)
        {
            data >> temp;
            if (temp > 9 || temp < 0)
            {
                cout << "数据非法" << endl;
                return -2;
            }
            a[i][j] = !temp * 10 + temp;
        }
    return 0;
}

void read_name(char name[][20], char file[])
{
    char file_[20];
    cout << "当前目录下符合sudoku*.txt的匹配文件有: " << endl;
    int num = filesearch(".", "sudoku*.txt", name);
    for (int i = 1; i < num; i++)
        cout << name[i] << "\t";
    cout << endl << "请输入数独题目文件名(回车表示默认sudoku.txt): ";
    char str = _getch();
    cout << str;
    if (str == '\r')
        strcpy(file, "sudoku.txt");
    else
    {

```

```
        cin >> file_;
        file[0] = str, file[1] = '\0';
        strcat(file, file_);
    }
}

bool read_name_(char name[][20], char file[])
{
    int num = filesearch(".", "sudoku*.txt", name), n = console_file(name, num + 1);
    if (n == -1)
        return 1;
    strcpy(file, name[n]);
    setcursor(hout, CURSOR_VISIBLE_NORMAL);
    return 0;
}
```

装

订

线