同济大学

高级语言程序设计课程实验报告

实验名称：汉诺塔综合演示实验

学　　院：计算机科学与技术学院

专　　业：计算机科学与技术

学　　号：2353761

姓　　名：王晢晶

完成日期：2024 年 11 月 21 日

1. 汉诺塔综合演示

1.1.题目描述

1.1.1. 题目目标

通过学习使用伪图形界面工具函数集，实现汉诺塔9个功能的综合。

1.1.2. 题目功能要求及函数使用限制

1. 除 总移动步数、圆柱上现有圆盘的编号、圆柱上现有圆盘的数量、延时 外，其余变量不允许定义为全局变量；

2. 功能5--在屏幕上画出三根圆柱、功能6--在起始圆柱上画出n个圆盘，需要加延时展示实现过程；

3. 功能7--第一次移动、功能8--汉诺塔演示过程，必须展示出先上移，再平移，后下移的移动过程；

4. 功能9--人工操作移动，需要检查输入的合理性以及判断是否符合移动规则，并给出相应的提示；

5. 延时速度不宜设置过大；

6. 1/2/3/4/8共用的递归函数不超过15行；

7. 1/2/3/4/6/7/8的输入必须共用一个函数；

8. 3/4/8必须共用一个横向输出打印的函数；

9. 4/8必须共用一个纵向输出函数；

10. 5/6/7/8/9必须共用一个画柱子的函数；

11. 7/8/9必须共用同一个控制盘子移动的函数；

12. 尽量每个函数不超过50行；

13. 必须VS2022编译通过；

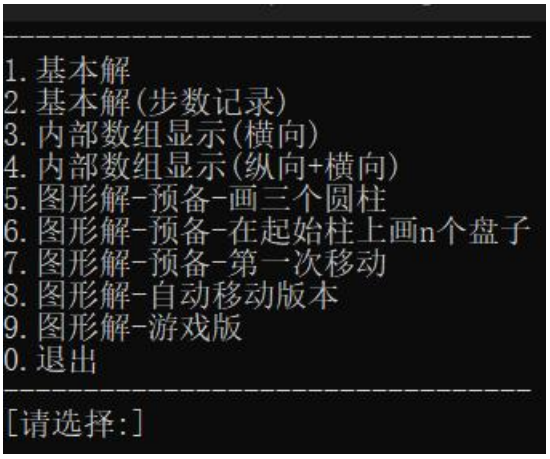
14. 对应的控制台及分辨率必须按照文档要求调整。

2. 整体设计思路

整个程序整体设计思路为：

首先调用hanoi_menu()函数，显示选择功能的界面，等待用户输入对应的功能号（由于

hanoi_menu()函数的返回值为对应功能号的ASCII码，所以后续对功能值的处理均为i-48）；



（功能对应如图，方便后续报告内容的理解）

如果输入的为0，则程序直接结束；如果不是0，首先经过**cct_getxy**(x,y)和**cct_gotoxy**(x,y)函数，使光标移动在合适的输出位置，等待后续的程序输出；

之后，均调用**shuru**(&src, &tmp, &dst, &layers,i-48)；**chushihua**(src, layers)；**set_sudu**(src, dst,layers,i-48)；**draw_column**(src,dst,i-48,layers)；**draw_original**(src,layers,i-48)；**chushihua2**(i - 48)；**hanoi**(layers, src, tmp, dst, i - 48)这些函数。显而易见，未必每个功能都需要经过这些函数处理，因此通过给各个函数增加gongnenghao这一参数，通过不同参数的传入，区分不同功能的实现。

（先对上述提到的函数（函数名的命名原则是英语和拼音混合，请注意辨别）的功能作以简单的说明，后续有对应的函数程序实现源代码：

shuru(&src, &tmp, &dst, &layers,i-48)--只有功能5不需要调用此函数,该函数主要是通过指针的使用，读取用户输入的起始柱、目标柱、层数，并传给main函数，便于后续函数的参数传递；

chushihua(src, layers)--由于在所有自定义函数的源程序文件中定义了几个此次作业允许定义的全局变量，即

```
int bushu = 1;      //记录移动步数
int a[10], b[10], c[10];  //存储盘号
int d, e, f;              //存储a，b，c柱的盘数
static int sudushijian;//移动延时的时间
```

每次实现对应功能前需要对这些全局变量进行初始化，以便后续程序正确地实现其他功能；

set_sudu(src, dst, layers,i-48)--只有功能4和8会调用该函数，主要是读取用户期望的实现延时移动速度，并对全局变量sudushijian进行赋值，然后清屏，准备下个函数的输出；

draw_column(src,dst,i-48,layers)--只有功能5/6/7/8/9会调用该函数，主要功能是画出三根圆柱，并显示清屏后第一行的“从 src 到 dst，共 layers 层”的字样；

draw_original(src,layers,i-48)--只有功能6/7/8/9会调用该函数，主要功能在对应的起始柱上画对应数量的圆盘；

chushihua2(i - 48)--只有4/8/9会调用该函数，主要功能是在竖向输出前显示各个柱子初始的状态；

hanoi(layers, src, tmp, dst, i - 48)--只有5/6/7/9不需要调用该函数，主要功能是通过递归解决汉诺塔每一步如何移动的问题）

后续由于功能7和9的实现需要另外的函数单独实现，故用switch-case语句对这两种情况单独执行，其余功能号直接跳过该语句块。

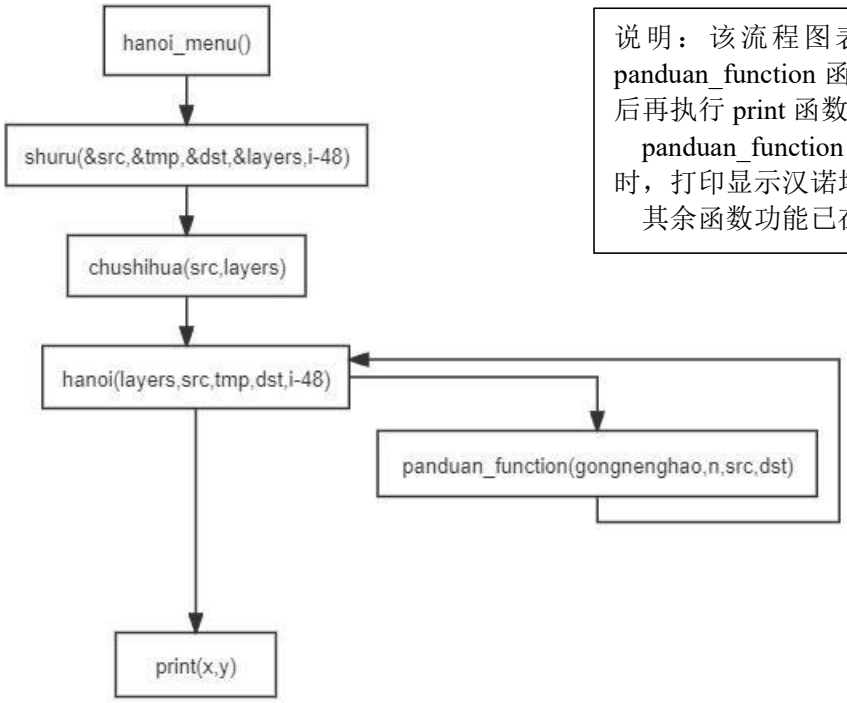
最后通过一个**print**（）函数，输出“按回车键继续”的字样，衔接下一次的功能选择。

3. 主要功能的实现

整个程序的功能实现描述已在整体设计思路中给出，此部分将以流程图的形式展示各个主要功能

的实现过程（即函数之间的相互调用关系）。

功能1：



说明：该流程图表示 hanoi 函数调用 panduan_function 函数，honoi 执行完之后再执行 print 函数。

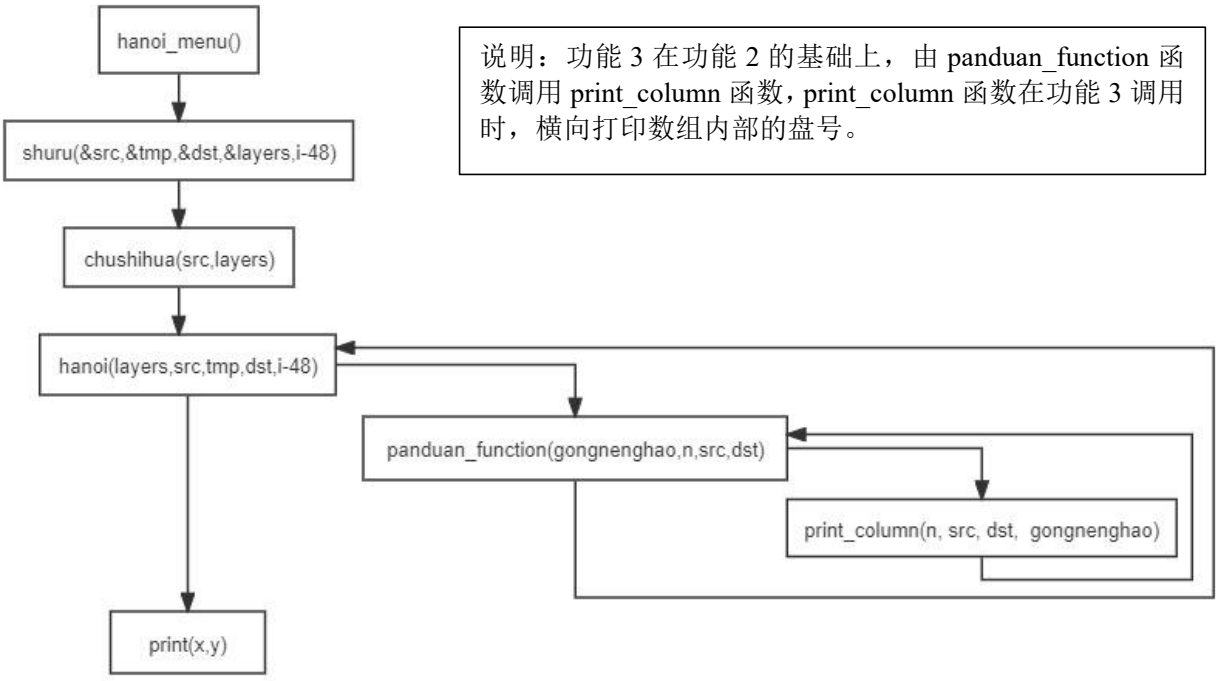
panduan_function 函数：在功能 1 调用时，打印显示汉诺塔的基本解。

其余函数功能已在前文提到。

功能2：

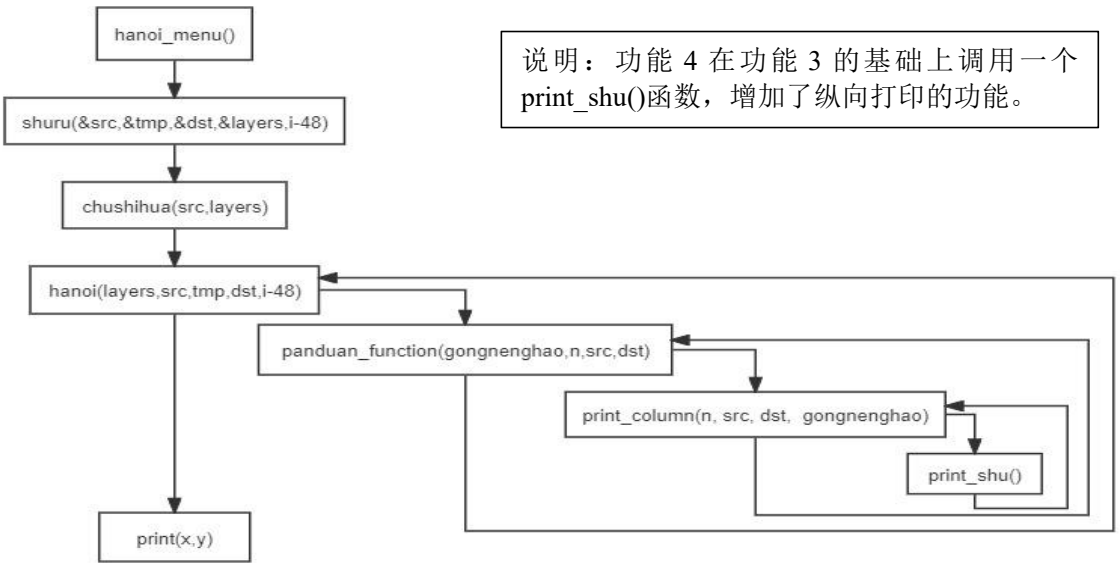
流程图和功能1相同，不同的是panduan_function的输出--在功能2调用时，该函数的功能是打印基本解，并显示步数记录。

功能3：



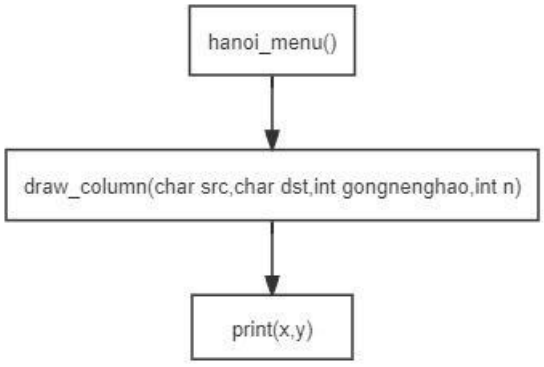
说明：功能 3 在功能 2 的基础上，由 panduan_function 函数调用 print_column 函数，print_column 函数在功能 3 调用时，横向打印数组内部的盘号。

功能4：

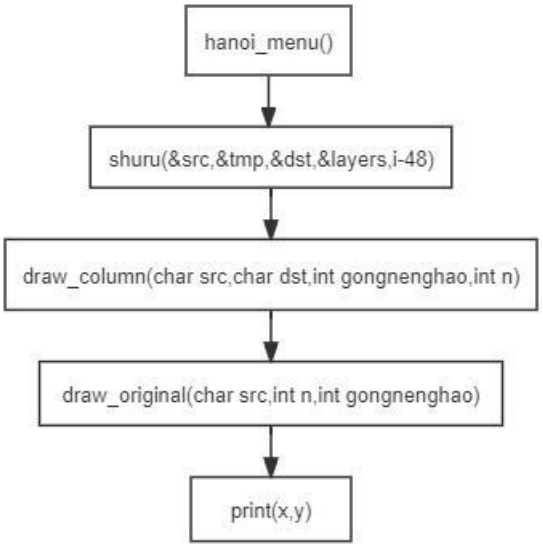


说明：功能 4 在功能 3 的基础上调用一个 print_shu()函数，增加了纵向打印的功能。

功能5：



功能6：



功能7：

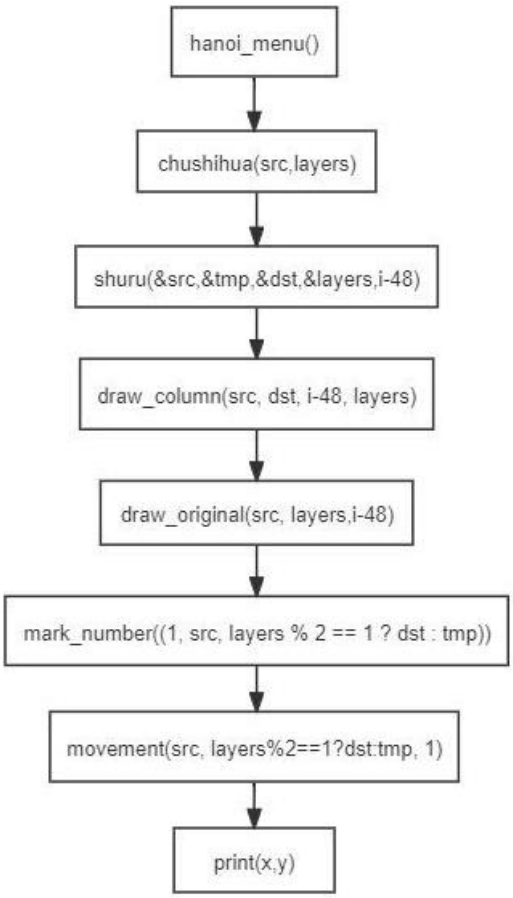
功能9：

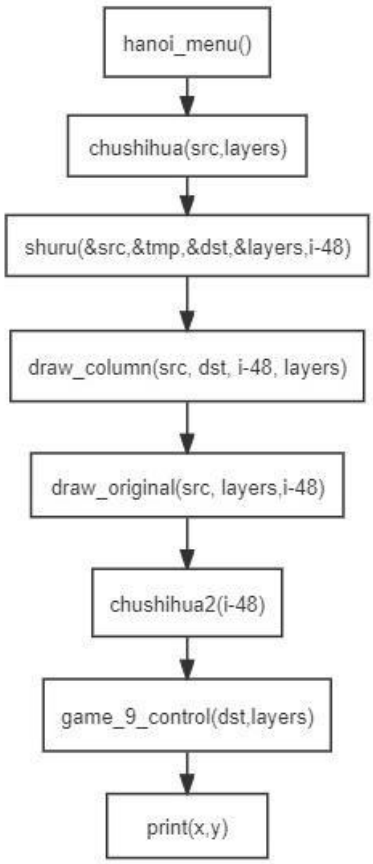
说明：功能 7 和功能 9 的一些函数功能已在前文说明，在此说明其他函数功能。

mark_number(int n, char src, char dst)--该函数主要功能是实现三个柱每次移动前后盘号、盘数等全局变量的更新；（其中功能 7 中，关于第一次移动的目标柱是通过层数的奇偶性来确定的）

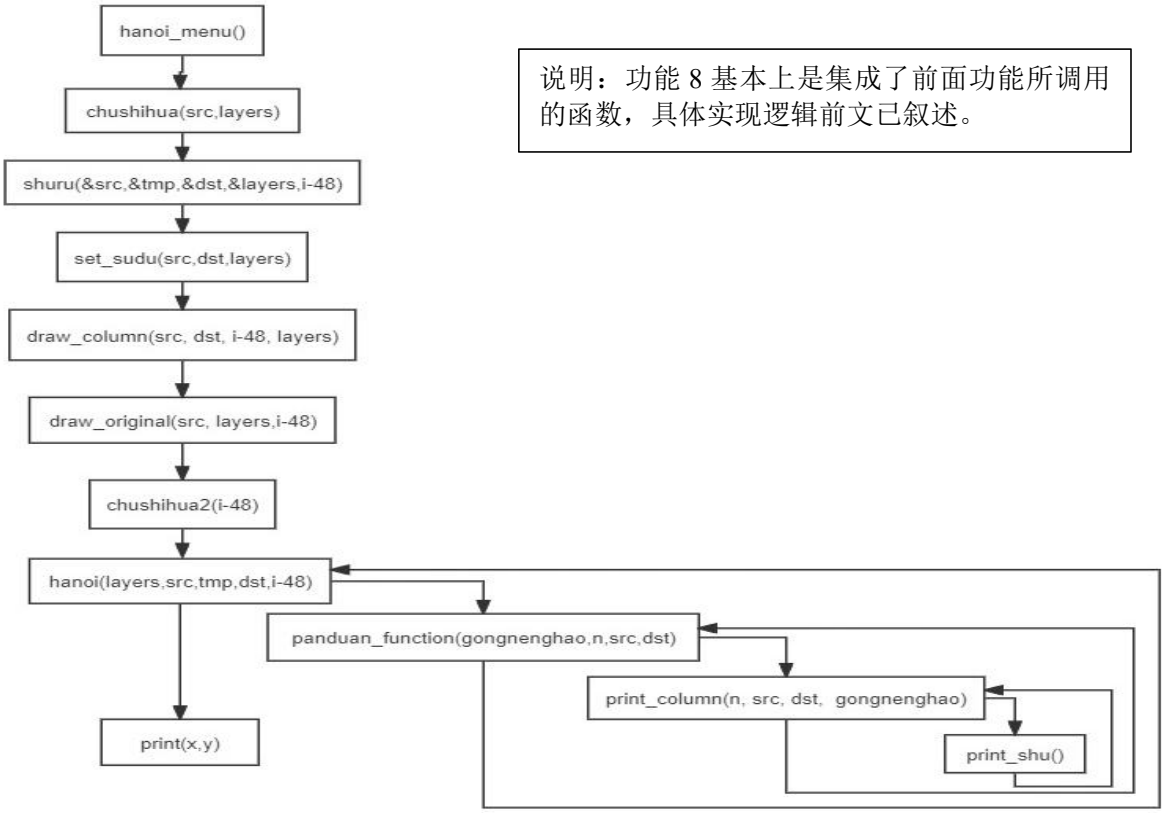
movement(char src, char dst, int n)--该函数功能则是通过调用相关的 cct 工具函数，实现圆柱上盘子移动的动画效果；

game_9_control (char dst,int layers)--该函数是专门为功能 9 设计，其调用了多个函数，具体的实现逻辑如图所示。



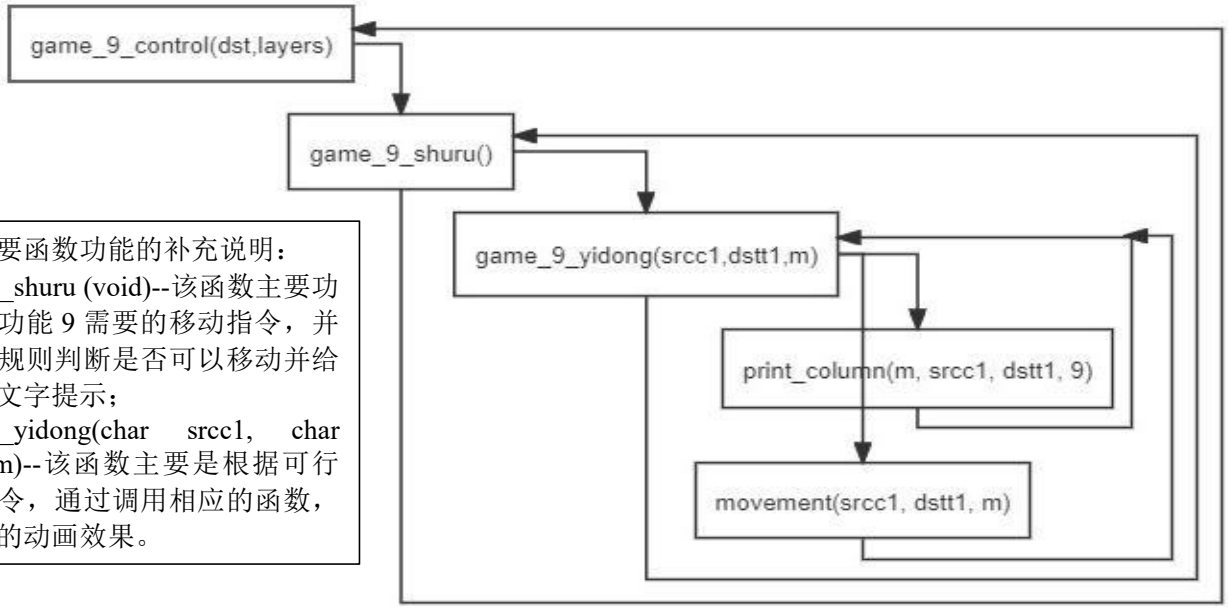


功能8：



说明：功能 8 基本上是集成了前面功能所调用的函数，具体实现逻辑前文已叙述。

附：game_9_control (char dst,int layers)的实现逻辑流程图



说明：主要函数功能的补充说明：
game_9_shuru (void)--该函数主要功能是接收功能 9 需要的移动指令，并根据移动规则判断是否可以移动并给出相应的文字提示；
game_9_yidong(char srcc1, char dstt1, int m)--该函数主要是根据可行的移动指令，通过调用相应的函数，实现移动的动画效果。

4. 调试过程碰到的问题

Question1描述:所有函数写完后进行功能9的验证，结果在验证过程中发现，明明初始时盘全在A柱，此时要将1号盘移向空柱B时，显示提示“不能大盘压小盘”，然后对对应的记录盘数的变量进行读取发现，明明B柱是空，却显示B柱中有1个盘子。

解决方案：经过对game_9_shuru()函数的仔细检查、逐步输出执行结果后发现，是数组读取越界了！

```
    break;
}//输入完成
int m, n;
if (srcc1 == 'A') {
    m = a[d - 1];
}
else if (srcc1 == 'B') {
    m = b[e - 1];
}
else {
    m = c[f - 1];
}
if (dstt1 == 'A') {
    n = a[d - 1];
}
else if (dstt1 == 'B')
    n = b[e - 1];
    cout << n;
    system("pause");
}
else {
    n = c[f - 1];
}
if (m == 0) {
    cout << endl;
    cout << "不要移动空的柱子";
    Sleep(1000);
```

红色方框内，当目标柱为空时，将索引为-1 的值赋给了 n，然后此时恰好索引值为-1 的存储空间内存的是 1，从而导致的出错。

```
if (srcc1 == 'A') {
    m =d==0?0: a[d - 1];
}
```

这样修改就可以避免越界读取的问题啦

与课内知识的联系：使用数组时一定要注意人为地保证数组索引值在合理范围内，因为 VS 编译过程不会对越界读取报错，就会导致错误很难查出来。

其他Question描述：其实在调试过程中遇到了很多问题，但检查过后发现无非就是由于函数本身逻辑

错误或者函数之间的调用关系有误，因为很容易检查出来并加以改正，因此没有在此叙述的意义。具体调试过程中的心得体会会在后文详细叙述。

5. 心得体会

5.1. 完成本次作业得到的一些心得体会，经验教训

其实完成此次汉诺塔综合实验的过程中能很明显地感觉到，要解决一道较为复杂地程序题时，最好是选择分为若干小题分别求解，但在处理整体函数之间的调用关系时，也要从整体的角度去思考。因此，各个函数执行逻辑的问题是局部的，要解决函数之间的相互调用关系还得从整体的角度去考虑。这算是对编程思想的一点点心得体会吧。所以在完成此次大作业的时候，我也是一边画流程框图，一边不断简化代码，修改函数调用关系，修改函数参数传递的。

然后就是一些编程技巧的总结。此次大作业的伪图形化实现，不可避免地涉及到了坐标的移动，比如柱子该画在哪里，纵向输出的又应该打印在哪里才不会干扰到画的柱子等等。此次大作业我采用了在源程序中写死坐标值的方法，缺点非常明显，就是在修改相对位置的时候改坐标改的非常痛苦，经常是改了这个函数里的，没改那个函数里的。当我经历了n次修改坐标的痛苦后，我深切地体会到了宏定义的好处。但由于一开始我并不是很习惯使用宏定义，所以一开始设计函数就是写死了坐标，等我觉得麻烦，想要改成宏定义的时候，程序已经写了很多，ddl也快到了，所以就没有修改成宏定义的方式。这算是一个非常大的经验教训，下次再遇到涉及坐标或相对位置的题目，我一定一定优先使用宏定义。

除此之外就是我对数组使用的总结。在使用数组时，务必定义的时候就将其索引值注释在旁边，每次引用数组的时候都要提醒自己注意索引值要在合理范围内，不然VS也不报错，出了错误真的很难查出来。

5.2. 在做一些较为复杂的程序时，是分为若干小题还是直接一道大题的形式更好?

分成若干小题更好，理由如上。

5.3. 汉诺塔完成了若干小题，总结你在完成过程中是否考虑了前后小题的关联关系，是否能尽可能做到后面小题有效利用前面小题已完成的代码，如何才能更好地重用代码?

在一开始完成过程中，我是考虑了在新写的函数中利用前面小题的相关代码的。具体方法是先不受长短的限制，只管调用，先写出能有效实现功能的代码，最后再根据长度的限制返回去简化代码。但毫无疑问，这样非常浪费时间。所以后来解决后面小题之前，我会先将已经写好的函数的定义写在纸上，旁边注明其能实现的功能，然后在写新函数的时候对照已有函数的功能进行编写，效率就会提高很多。

5.4. 以本次作业为例，说明如何才能更好地利用函数来编写复杂的程序

我认为利用函数来编写复杂程序的关键就是提高函数的调用率。因此编写程序的时候需要记录已写好的函数需要传递哪些参数，实现哪些功能等，这样才能在编写新的程序时提高函数的利用率，降低代码量。但相应的可能逻辑会复杂很多，注意厘清即可。

6. 附件：源程序

```
int  hanoi_menu (void); //函数声明里必须加上void，否则报错
void shuru(char* src, char* tmp, char* dst,int*n,int gongnenghao);
void hanoi(int n, char src, char tmp, char dst, int gongnenghao);
void print(int x, int y);
void draw_column(char src,char dst,int gongnenghao,int n);
void draw_original(char src,int n,int gongnenghao);
void print_column(int n, char src, char dst, int gongnenghao);
void chushihua(char src, int n);
void movement(char src, char dst, int n);
void set_sudu(char src, char dst, int layers,int gongnenghao);
void mark_number(int n, char src, char dst);
int game_9_shuru (void);
void game_9_control (char dst,int layers);
void print_title(int gongnenghao, char src, char dst, int n);
void print_shu();
void yanshipanduan(int gongnenghao, char src, char dst, int n);
void panduan_function(int gongnenghao, int n, char src, char dst);
void chushihua2 (int gongnenghao);
void game_9_yidong(char srcc1, char dstt1, int m);
```

本部分将给出上图框起来的函数源程序，未给出的为常规输入处理或输出文字的常规内容。

```
void mark_number(int n,char src, char dst)
{
    int i, j;
    if (src == 'A') {
        i = d;
        a[i - 1] = 0;
        --d;
    }
    else if (src == 'B') {
        i = e;
        b[i - 1] = 0;
        --e;
    }
    else {
        i = f;
        c[i - 1] = 0;
        --f;
    }

    if (dst == 'A') {
        j = d;
        a[j] = n;
        ++d;
    }
    else if (dst == 'B') {
        j = e;
```

```
            b[j] = n;
            ++e;
        }
        else {
            j = f;
            c[j] = n;
            ++f;
        }
        return;

}

void print_column(int n, char src, char dst,int gongnenghao)
{
        mark_number(n, src, dst);
        if (gongnenghao == 2) {
            cout << endl;
        }
        else {
            cout << "A:";
            for (int m = 0; m < 10; ++m) {
                if (a[m] > 0) {
                    cout << setw(2) << a[m];
                }
                else {
                    cout << "  ";
                }
            }
            cout << " ";

            cout << "B:";
            for (int m = 0; m < 10; ++m) {
                if (b[m] > 0) {
                    cout << setw(2) << b[m];
                }
                else {
                    cout << "  ";
                }
            }
            cout << " ";

            cout << "C:";
            for (int m = 0; m < 10; ++m) {
                if (c[m] > 0) {
                    cout << setw(2) << c[m];
                }
                else {
                    cout << "  ";
                }
            }
            cout << " ";
            cout << endl;
            if (gongnenghao != 3) {
                print_shu();
            }
        }
        return;
```

```
}

void print_shu()//纵向输出函数
{
    int x = 11, y = 32;
    for (int m = 0; m < 10; ++m) {
        cct_gotoxy(x, y);
        if (a[m] > 0) {
            cout << a[m];
            --y;
        }
        else {
            cout << " ";
        }
    }
    x = x + 10;
    y = 32;
    for (int m = 0; m < 10; ++m) {
        cct_gotoxy(x, y);
        if (b[m] > 0) {
            cout << b[m];
            --y;
        }
        else {
            cout << " ";
        }
    }
    x = x + 10;
    y = 32;
    for (int m = 0; m < 10; ++m) {
        cct_gotoxy(x, y);
        if (c[m] > 0) {
            cout << c[m];
            --y;
        }
        else {
            cout << " ";
        }
    }
    return;

}

void draw_column(char src,char dst,int gongnenghao,int n)
{
    if ( gongnenghao == 5 || gongnenghao == 6 || gongnenghao == 7 ||
        gongnenghao == 8 || gongnenghao == 9) {
        if (gongnenghao != 8) {
            print_title(gongnenghao, src, dst, n);
        }
        int x;
        int y;
        cct_setcursor(CURSOR_INVISIBLE);
        /*先画三个底柱*/
        x = 1;
        y = 20;
        /* 在坐标(1,20)位置处打印第一个底柱 */
```

```
            cct_showch(x, y, ch, bg_color, fg_color, LENGTH);
            x = 1 + LENGTH + SPACE;                                          //A 柱子起始 x：1-24
            cct_showch(x, y, ch, bg_color, fg_color, LENGTH);
            x = 1 + (LENGTH + SPACE) * 2;                                    //B 柱起始 x：33-56
            cct_showch(x, y, ch, bg_color, fg_color, LENGTH);                //C 柱起始 x: 65-88
            /*开始打印条柱*/
            for (y = 19; y >= 5; --y) {
                x = 1 + (LENGTH - 1) / 2;
                cct_showch(x, y, ch, bg_color, fg_color, 1);
                Sleep(30);
                x = 1 + LENGTH + SPACE + (LENGTH - 1) / 2;
                cct_showch(x, y, ch, bg_color, fg_color, 1);
                Sleep(30);
                x = 1 + (LENGTH + SPACE) * 2 + (LENGTH - 1) / 2;
                cct_showch(x, y, ch, bg_color, fg_color, 1);
                Sleep(30);
            }
            //增设常量 A 柱中间值 x1=1+(LENGTH-1)/2；B 柱中间值 x2=1+LENGTH+SPACE+(LENGTH-1)/2;C 柱中间值
x3=1+(LENGTH+SPACE)*2+(LENGTH-1)/2
            cct_setcolor(COLOR_BLACK, COLOR_WHITE);
            cct_setcursor(CURSOR_VISIBLE_NORMAL);
        }
        return;

    }

    void draw_original(char src,int n,int gongnenghao)
    {
        if (gongnenghao == 6 || gongnenghao == 7 || gongnenghao == 8 || gongnenghao == 9) {
            cct_setcursor(CURSOR_INVISIBLE);

            int x, y = 19;
            if (src == 'A') {
                x = 1 + (LENGTH - 1) / 2;
            }
            else if (src == 'B') {
                x = 1 + LENGTH + SPACE + (LENGTH - 1) / 2;
            }
            else {
                x = 1 + (LENGTH + SPACE) * 2 + (LENGTH - 1) / 2;
            }
            for (int h = n; h >= 1; --h) {
                cct_showch(x - h, y, ' ', h, fg_color, 2 * h + 1);
                --y;
                Sleep(30);
            }
            cct_setcolor(COLOR_BLACK, COLOR_WHITE);
            cct_setcursor(CURSOR_VISIBLE_NORMAL);
        }
        return;

    }

    void yanshipanduan(int gongnenghao, char src, char dst, int n)
    {
        if (gongnenghao == 4 || gongnenghao == 8) {
```

```
            if (gongnenghao == 8) {
                movement(src, dst, n);
            }
            if (sudushijian) {
                Sleep(sudushijian);
            }
            else {
                int huiche;
                while (1) {
                    huiche = _getch();
                    if (huiche == 13) {
                        break;
                    }
                }
            }
        }
        return;

    }

    void panduan_function(int gongnenghao, int n, char src, char dst)
    {
        if (gongnenghao == 1) {
            cout << n << '#' << ' ' << src << '-' << '-' << '>' << dst << endl;
        }
        else {
            if (gongnenghao != 2&&gongnenghao!=3) {
                cct_gotoxy(11, 36);
            }
            cout << "第" << setw(4) << bushu << " 步(" << setw(2) << n << "#: " << src << "-->" << dst << ")
";
            print_column(n, src, dst, gongnenghao);
            yanshipanduan(gongnenghao, src, dst, n);
        }
        return;

    }

    void hanoi(int n, char src, char tmp, char dst, int gongnenghao)  //n 是盘号 从上到下为 1，2，3...传入
功能号
    {
        if (gongnenghao != 5 && gongnenghao != 6 && gongnenghao != 7 && gongnenghao != 9) {
            if (n == 1) {
                panduan_function(gongnenghao, n, src, dst);
                ++bushu;
                return;
            }
            hanoi(n - 1, src, dst, tmp, gongnenghao);
            panduan_function(gongnenghao, n, src, dst);
            ++bushu;
            hanoi(n - 1, tmp, src, dst, gongnenghao);
        }
        return;

    }
```

```
void movement(char src, char dst, int n)
{
    cct_setcursor(CURSOR_INVISIBLE);
    int srcc, dstt, x1, x2;
    if (src == 'A') {
        srcc = d;
        x1 = 1 + (LENGTH - 1) / 2;
    }
    else if (src == 'B') {
        srcc = e;
        x1 = 1 + LENGTH + SPACE + (LENGTH - 1) / 2;
    }
    else {
        srcc = f;
        x1 = 1 + (LENGTH + SPACE) * 2 + (LENGTH - 1) / 2;
    }

    if (dst == 'A') {
        dstt = d;
        x2 = 1 + (LENGTH - 1) / 2;
    }
    else if (dst == 'B') {
        dstt = e;
        x2 = 1 + LENGTH + SPACE + (LENGTH - 1) / 2;
    }
    else {
        dstt = f;
        x2 = 1 + (LENGTH + SPACE) * 2 + (LENGTH - 1) / 2;
    }
    //上移
    for (int y = 19 - srcc; y >= Y; y--) {
        cct_showstr(x1-n, y, "  ", n, fg_color, 2 * n+1 );
        Sleep(sudushijian+20);
        if (y >= Y+1) {
            cct_showch(x1-n, y, ' ', COLOR_BLACK, COLOR_WHITE, 2 * n+1);
        }
        if (y >= 5) {
            cct_showch(x1, y, ' ', COLOR_HYELLOW, COLOR_WHITE, 1);
        }
    }
    if (x1 < x2) {  //右移(x1<x2)
        for (int x = x1; x <= x2; x++) {
            cct_showch(x-n, Y, ' ', n, fg_color, 2 * n+1);
            Sleep(sudushijian+20);
            if (x <= x2 - 1) {
                cct_showch(x-n, Y, ' ', COLOR_BLACK, COLOR_WHITE, 2 * n + 1);
            }
        }
    }
    else {     //左移 x1>x2
        for (int x = x1; x >= x2; x--) {
            cct_showch(x-n, Y, ' ', n, fg_color, 2 * n + 1);
            Sleep(sudushijian+20);
            if (x >= x2 - 1) {
                cct_showch(x-n, Y, ' ', COLOR_BLACK, COLOR_WHITE, 2 * n + 1);
            }
        }
```

```
        }
        //下移
        for (int y = Y; y <= 19-dstt+1; y++) {
            cct_showstr(x2-n, y, " ", n, fg_color, 2*n + 1);
            Sleep(sudushijian+20);
            if (y <= 19-dstt) {
                cct_showch(x2-n, y, ' ', COLOR_BLACK, COLOR_WHITE, 2 * n + 1);
            }
            if (y >= 5&&y<=19-dstt) {
                cct_showch(x2, y, ' ', COLOR_HYELLOW, COLOR_WHITE, 1);
            }
        }
        cct_setcolor(COLOR_BLACK,COLOR_WHITE); //恢复缺省颜色--黑底白字
        cct_setcursor(CURSOR_VISIBLE_NORMAL);
        return;

    }

    void game_9_yidong(char srcc1, char dstt1,int m)
    {
        cct_gotoxy(9, 33);
        cout << "=========================" << endl;
        cct_gotoxy(9, 34);
        cout << "  A         B         C" << endl;
        cct_gotoxy(11, 36);
        cout << "第" << setw(4) << bushu << " 步(" << setw(2) << m << "#: " << srcc1 << "-->" << dstt1 << ")
";
        ++bushu;
        print_column(m, srcc1, dstt1, 9);
        movement(srcc1, dstt1, m);

    }

    void game_9_control(char dst,int layers)
    {
        int i=1;
        if (dst == 'A') {
            while (d != layers&&i==1) {
                cct_gotoxy(0, 38);
                i=game_9_shuru();
            }
        }
        else if (dst == 'B') {
            while (e!= layers && i == 1) {
                cct_gotoxy(0, 38);
                i=game_9_shuru();
            }
        }
        else {
            while (f != layers && i == 1) {
                cct_gotoxy(0, 38);
                i=game_9_shuru();
            }
        }
        if (i == 0) {
            cct_gotoxy(0, 39);
```

```
            cout << "再来一局！";
            return;
        }
        cct_gotoxy(0, 39);
        cout << "游戏结束，太棒啦！！";
        return;


}
```