### 大作业——汉诺塔实验报告

计1 1751151 郭思远

完成日期: 12月19日

#### 1. 汉诺塔

输入起始柱,目标住,和圆盘数量(1-10),起始柱上从小到大有n个原盘,需要按大小顺序重新摆放在另一根柱子上。并且规定,大圆盘始终不能压小圆盘,在三根柱子之间一次只能移动一个圆盘,求移动的最少步数。

#### 1.1基本解+基本解(步数记录)

给出最少步数每一步的移动方向和移动盘号,并记录步数。

#### 1.2内部数组显示(横向+纵向)

要求给出移动过程中每根柱子上现有的圆盘数量及编号。显示方式为横式加竖式。

#### 1.3图形解预备

订

线

在屏幕上画出三根圆柱,从左到右编号为A、B、C,输入起始柱和圆盘数量(1-10),在起始柱上从小到大画n个盘,盘子颜色各不相同。并画出第一次移动的动画,要求从起始柱上移出,平移后下落到目标柱。

### 1.4图形解自动移动版

自动完成全部最小步数移动过程,移动方式必须是上移,平移,下移。

### 1.5游戏版汉诺塔(人工操作移动步骤)

键盘输入两个字母代表本次移动的源和目标,移动时检查合理性(大盘压小盘,源柱为空提示出错,并重输),合理移动记录步数。移动必须上移,平移,下移,待所有按序移动到结束柱则提示"游戏结束"。

### 2. 整体设计思路

使用三个全局一维数组来显示内部数组和三个全局变量atop, btop, ctop记录数组的栈顶+1。

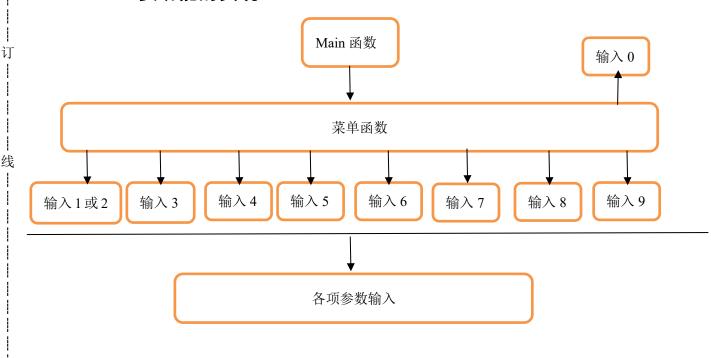
整个程序的核心是递归函数,递归函数需实现将n-1个盘子移到中间柱,将第n个盘子移到目标柱,再将n-1个盘子移到目标柱。

用三个一维数组模拟进出栈的情况。

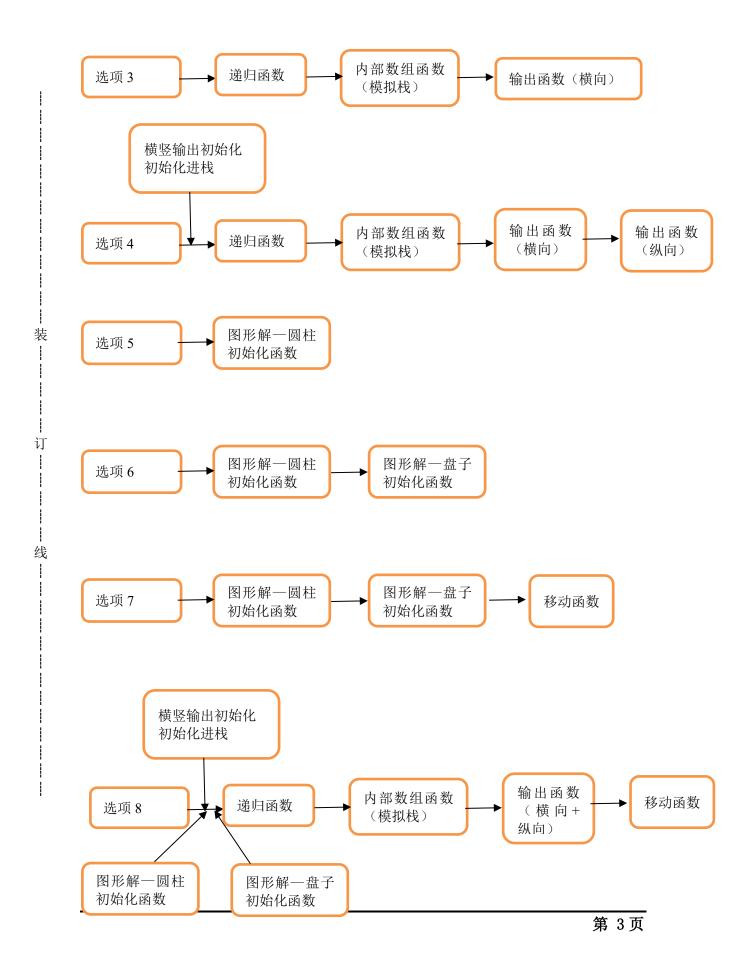
在递归函数中调用模拟栈的函数,在模拟栈的函数中加入输出函数,输出内部数组。

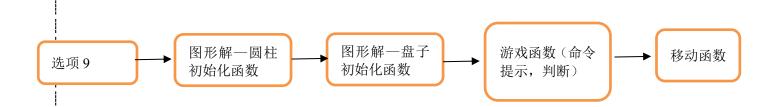
每一个主要功能由一个函数完成,其他函数调用该函数完成一个选项的要求。

#### 3. 主要功能的实现









#### 4. 调试过程碰到的问题

#### 4. 1.

装

订

线

显示内部数组时(竖式),在打完数组后会接着打印0出来。一开始想的解决办法是输出一次清屏一次,但是闪得特别厉害而且整体感觉很丑。于是想到从美观角度来说应该只屏蔽零的位置,于是通过每一次从top位开始到10结束输出""进行消除。

#### 4. 2.

在图形解移动函数调试的过程中,暴露了很多写第一次移动没考虑到的问题。移动色块颜色混乱,消除函数没有完全消除,错位,盘子不能完全落下去等等问题。一开始想暴力写,定死每个盘子的颜色,大小,后想一想觉得太冗长了。又想到可以通过盘号h这个参数去固定颜色和大小,用表示栈顶的全局变量去控制下落位置,使得代码较为精简。

#### 4, 3,

选择一个选项,执行完毕后按回车键返回菜单,再选择下一个选项执行时发现步数记录没有清零。一开始想去掉静态变量,改为传参记录,失败(并且觉得方法不好)。后来又想到可以一开始记录下目标柱,通过表示栈顶的全局变量等于盘子总数判断递归结束,然后把静态变量清零。后来发现可以用外置开关方法,传入一个参数,该参数为零就把这个静态变量清零,修改较少,而且比较精简。

### 5. 心得体会

#### 5.1. 完成本次作业的心得体会, 经验教训

之前没有精简代码使得我非常痛心和后悔,花了不少时间把之前的烂代码改得好看了 许多。多了以后每周四晚上把上周作业进行一定修改。

每一个主要功能用一个函数输出,在查找问题时效果很好,改bug的时间明显缩短。并且借此作业学会了之前一直没太弄明白的debug,确实很好用。

在写伪图形界面时,对于整体布局一开始没有考虑周全,导致排版很丑,希望以后吸

取教训。

订

线

# 5.2. 在做一些较为复杂的程序时,是分成若干小题好,还是直接一道大题好?

分成若干小题好。可以比较快地上手,难度会因为分解而下降,主观上也更愿意去不断 尝试解决问题。 随着问题不断加深,解题思路也有一种连贯性和整体感。

#### 5.3. 前后小题的关联性问题.

总体上注意到了前后小题的关联性。

更好重用代码的方法是,代码按照功能归类成为一个函数模板。通过修改利用模板使得 更好重用代码。

#### 5.4. 如何更好地利用函数来编写复杂程序.

```
using namespace std;
void CD();//菜单
int get_int(char *prompt, int min_value, int max_value, char *p1, char *p2);//输入
void hengchushihua();//模式初始化
void shuchushihua();//竖式初始化
void CSHzhuzi();//初始化图形柱子
void CSHpanzi(int n, char qsz);//初始化盘子图形
void DG(int n, char one, char two, char three, int sleep, int choice, int t);//递归函数
int hanoidg(char from, char to, int sleep, int n, int choice, int t);//内部数组
void go(char from, char to, int h, int num, int choice);//横向输出
void shu();//纵向输出
void move(int n, char qsz, char mbz, int h, int choice);//移动
void game(int n, char qsz, char mbz, int choice);//游戏版
```

面对复杂程序,先将其分解开来。汉诺塔大作业主要的实现功能分解为菜单、输入、递归、内部数组、横纵输出、移动、游戏版,将这些重要功能列出作为函数,不仅将题目细化拆分使得容易入手,而且使得查找bug,修改程序更加快速、方便。

另外由于初始化需要的次数也比较多,可以拿出来作为一个函数,使得代码精简,整体有一种整洁感。

对于常见的功能和模板要整理成为模板,比如输入函数get\_int.在以后面对其他复杂程序时可以套用模板,减少时间消耗。

第 5 页

#### 6. 源程序

装

订

```
/*1751151 计1 郭思远*/
                                                                                      a[0] = a[0] - 32;
#define _CRT_SECURE_NO_WARNINGS
                                                                                      a[1] = a[1] - 32;
#include<iostream>
                                                                                if ((a[0] >= 'A' \&\& a[0] <=
#include <cstdio>
                                                                'C'&&a[1] >= 'A'&&a[1] <= 'C' && (a[0] != a[1]))
#include<windows.h>
#include<iomanip>
                                                                || (a[0] = 'Q'))
#include <comio.h>
#include<cmath>
                                                                                      *p1 = a[0];
                                                                                      p2 = a[1];
#include "cmd_console_tools.h"
using namespace std;
                                                                                      break;
void CD();//菜单
int get_int(char *prompt, int min_value, int
max_value, char *p1, char *p2);//输入
                                                                           return 0;
void hengchushihua();//横式初始化
void shuchushihua();//竖式初始化
                                                                     if (min_value == 'A')
void CSHzhuzi();//初始化图形柱子
void CSHpanzi(int n, char qsz);//初始化盘子图形
                                                                           while (1)
void DG(int n, char one, char two, char three,
                                                                                cout << prompt << "(" <<
int sleep, int choice, int t);//递归函数
                                                                (char)min_value << "-" << (char)max_value << ")"
int hanoidg(char from, char to, int sleep, int
n, int choice, int t);//内部数组
                                                                << end1;</pre>
                                                                                cin >> value2;
void go (char from, char to, int h, int num, int
                                                                                if (value2 == 'a' || value2 ==
choice);//横向输出
void shu();//纵向输出
                                                                'b' || value2 == 'c')
void move(int n, char qsz, char mbz, int h, int
choice);//移动
                                                                                      value2 = value2 - 32:
void game(int n, char qsz, char mbz, int
                                                                                      break;
choice);//游戏版
                                                                                else if (value2 == 'A' ||
void setcursor(const HANDLE hout, const int
                                                               value2 == 'B' || value2 == 'C')
options):
#define N 10
                                                                                      break;
int a[N], b[N], c[N];
int atop = 0, btop = 0, ctop = 0;
                                                                           return value2;
int main()
                                                                     }
{
                                                                     else
     CD();
     HANDLE hout =
                                                                           while (1)
GetStdHandle(STD_OUTPUT_HANDLE);
     gotoxy(hout, 0, 20);
                                                                                if (max_value == 9)
     return 0;
                                                                                      gotoxy(hout, 0, 13);
int get_int(char *prompt, int min_value, int
                                                                                      cout << '
                                                               " ;
max_value, char *p1, char *p2)//输入
                                                                                      gotoxy (hout, 0, 13);
     const int BASE X = 10;
                                                                                      cout << prompt;</pre>
     const int BASE_Y = 28;
     HANDLE hout =
                                                                                else
GetStdHandle(STD_OUTPUT_HANDLE);
     char a[3];
                                                                                      cout << prompt << end1;</pre>
     int value1;
                                                                                cin >> value1;
     char value2;
     if (min_value == 'a')
                                                                                if (!cin.good())
           while (1)
                                                                                      cin.clear();
                                                                     cin.ignore(cin.rdbuf()->in_avail(), '\n');
                gotoxy(hout, BASE_X, BASE_Y +
14);
                                                                                      continue;
                cout << "
" << end1;
                                                                                if (value1 >= min_value&&value1
                gotoxy(hout, BASE_X, BASE_Y +
                                                               <= max value)
14);
                                                                                      break:
                cout << prompt;</pre>
                cin >> a;
                                                                           return value1;
                if (a[0] >= 'a' \&\& a[0] <=
'z'&&a[1] >= 'a'&&a[1] <= 'z')
                                                               void CSHzhuzi()//初始化图形柱子
```

装

订

```
{
                                                              'A', 'C', 0, 0);
     int i, j;
                                                                              while (1)
     HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
                                                                                   mbz = get_int("请输入目标
     const int BASE_X = 10;
                                                              柱", 'A', 'C', 0, 0);
     const int BASE_Y = 28;
                                                                                   if (mbz == qsz)
     const int bg_color = COLOR_HYELLOW;
                                                                                        continue;
     const int fg_color = COLOR_WHITE;
                                                                                   else
     const char ch = '';
                                                                                        break;
     for (i = 0; i < 3; i++)
                                                                              z jz = 'A' + 'B' + 'C' - mbz -
          showch (hout, BASE X + i * 36,
                                                             asz:
BASE_Y, ch, bg_color, fg_color, 26);
          showch (hout, (BASE X + 26) + 36 * i,
                                                                              if (qsz == 'A')
BASE_Y, ch, COLOR_BLACK, fg_color, 10);
                                                                                   atop = n;
                                                                              if (qsz == 'B')
     for (j = 0; j < 15; j++)
                                                                                  btop = n;
                                                                              if (qsz == 'C')
          showch (hout, BASE_X + 13, BASE_Y -
                                                                                   ctop = n;
j, ch, bg_color, fg_color, 1);
          showch (hout, BASE_X + 13 + 1 \star 36,
                                                                              for (i = 0; i < N; i++)//初始化
BASE_Y - j, ch, bg_color, fg_color, 1);
                                                                                   a[i] = b[i] = c[i] = 0;
          showch (hout, BASE_X + 13 + 2 * 36,
                                                                              for (i = 0; i < n; i++)//进栈
BASE_Y - j, ch, bg_color, fg_color, 1);
                                                                                   if (qsz == 'A')
          Sleep (100);
                                                                                       a[i] = n - i;
     showch (hout, 100, BASE_Y + 100, '',
                                                                                   else if (qsz = 'B')
                                                                                       b[i] = n - i;
COLOR_BLACK, COLOR_WHITE, 1);
                                                                                   else if (qsz = 'C')
void CD()
                                                                                        c[i] = n - i;
     int num = 1;
                                                                        if (choice == 4 \mid \mid choice == 8)
     int choice;
     char gsy;
                                                                              sleep = get_int("请输入移动速度
     cout <<
                                                              (0-5: 0-按回车单步演示 1-延时最长 5-延时最短)",
end1:
     cout << "1、基本解" << endl;
                                                             0, 5, 0, 0);
     cout << "2、基本解(步数记录)" << endl;
cout << "3、内部数组显示 (横向)" << endl;
                                                                        setcursor(hout, CURSOR_INVISIBLE);
     cout << "4、内部数组显示(纵向+横向)" <<
                                                                        setconsoleborder (hout, 120, 50);
end1;
                                                                        if (choice == 4 || choice == 8 ||
     cout << "5、图形解-预备-画三个圆柱" <<
                                                             choice == 9)
end1;
                                                                              hengchushihua();
     cout << "6、图形解一预备一在起始柱上画n个
                                                                        if (choice == 4 || choice == 8 ||
盘子" << endl;
                                                             choice == 9)
     cout << "7、图形解—预备—第一次移动" <<
                                                                              shuchushihua();
                                                                        if (choice \geq= 5)
end1:
     cout << "8、图形解一自动移动版本" << end1;
     cout << "9、图形解—游戏版" << endl;
                                                                              CSHzhuzi():
     cout << "0、退出" << endl;
                                                                              if (choice == 5)
     cout << "-
                                                                              {
end1;
                                                                                   gotoxy(hout, 0,36);
     choice = get_int("请选择[0-9]", 0, 9, 0,
                                                                                   cout << "按回车键继续" <<
0);
                                                             end1;
     if (choice != 0)
                                                                                   gsy = _getch();
                                                                                   system("cls");
          HANDLE hout =
                                                                                   CD();
                                                                              }
GetStdHandle(STD_OUTPUT_HANDLE);
          int n, i = 0;
                                                                        if (choice >= 6)
          int sleep;
          char qsz, mbz, zjz;
          const int BASE_X = 10;
                                                                              CSHpanzi(n, qsz);
          const int BASE_Y = 40;
                                                                              if (choice == 6)
          if (choice != 5)
                                                                                   gotoxy(hout, 0, 36);
                                                                                   cout << "按回车键继续" <<
                n = get_int("请输入塔的层数(1-
10):", 1, 10, 0, 0);
                                                             end1;
                qsz = get_int("请输入起始柱",
                                                                                   gsy = _getch();
```

装

订

```
system("cls");
                                                                           m = atop;
                                                                      if (qsz = 'B')
                      CD();
                                                                          m = btop;
                                                                      if (qsz = 'C')
          }
if (choice == 7)
                                                                           m = ctop;
   move(n, qsz, mbz, 1, choice);
                                                                      if (choice == 7)
if (choice != 9 && choice != 5 && choice != 6 &&
                                                                           /* 将第一个盘子从下向上移动 */
choice != 7)
                                                                           for (y = 0; y < 19 - n; y++)
if (choice = 4 \mid | choice = 8)
     DG(n, qsz, zjz, mbz, sleep, choice, 1);
                                                                                 showch (hout, BASE_X + 9 + (qsz)
                                                                - 'A') * 36 + 2, BASE Y - n - y, '
     DG(n, qsz, zjz, mbz, 6, choice, 1);
                                                                fg_color, 5);
                                                                                 Sleep (100);
if (choice == 4 \mid | choice == 8)
                                                                                 if (y < 15 - n)
     DG(n, qsz, zjz, mbz, sleep, choice, 0);
                                                                      showch (hout, BASE_X + 9 + (qsz - 'A') * 36
     DG(n, qsz, zjz, mbz, 6, choice, 0);
if(choice==8||choice==4)
                                                                + 4, BASE_Y - n - y, ' ', COLOR_HYELLOW,
                                                                COLOR_WHITE, 1);
     gotoxy(hout, 0, 45);
cout << "按回车键继续" << end1;
                                                                                      showch (hout, BASE_X + 9 +
                                                                (qsz - A') * 36 + 2, BASE_Y - n - y,
     gsy = _getch();
     system("cls");
                                                                COLOR_BLACK, COLOR_WHITE, 2);
     CD();
                                                                                      showch (hout, BASE X + 9 +
                                                                (qsz - A') * 36 + 5, BASE_Y - n - y, '
                                                                COLOR_BLACK, COLOR_WHITE, 2);
           if (choice == 9)
                 game(n, qsz, mbz, choice);
                                                                                 if (y >= 15 - n \&\& y < 18 - n)
                                                                                      showch(hout, BASE_X + 9 +
                 gotoxy (hout, 0, 45);
                 cout << "按回车键继续" << end1;
                                                                (qsz - A') * 36 + 2, BASE_Y - n - y,
                 gsy = getch();
                                                                COLOR_BLACK, COLOR_WHITE, 5);
                 system("cls");
                CD();
                                                                           /* 将第一个盘子从当前柱子平移到另一
                                                                根柱子 */
                                                                for (x = 0; x < 37; x++)
                                                                      if (qsz != 'B')
void CSHpanzi(int n, char qsz)//初始化盘子图形
                                                                       showch (hout, BASE_X + 9 + (qsz - 'A') *
     int i, j = 1;
     HANDLE hout =
                                                                36 + 2 + x*('A' - qsz + 1), BASE_Y - n - y + 1,
GetStdHandle(STD OUTPUT HANDLE);
                                                                 ', 3, fg_color, 5);
     const int BASE_X = 10;
                                                                     Sleep (100);
     const int BASE Y = 28;
                                                                    if (x < 36)
     const int fg_color = COLOR_WHITE;
const char ch = ' ';
                                                                        showch (hout, BASE_X + 9 + (qsz - 'A') *
                                                                36 + 2 + x*('A' - qsz + 1), BASE_Y - n - y + 1,
                                                                 ', COLOR_BLACK, COLOR_WHITE, 5);
     for (i = 10; i >= 1; i--)
                                                                                if (qsz == 'B') {
           \text{if } (n == i) \\
                                                                showch (hout, BASE_X + 9 + (qsz - 'A') * 36 + 2 +
           {
                                                                x*('A' - qsz), BASE_Y - n - y + 1, '', 3,
                showch (hout, BASE X + (12 - n)
+ (qsz - 'A') * 36, BASE_Y - j, ch, 2 + i,
                                                                fg_color, 5);
fg color, 25 - 2 * (11 - n);
                                                                Sleep (100);
                j++;
                                                                if (x < 36)
                                                                   showch (hout, BASE_X + 9 + (qsz - 'A') * 36 +
                n--:
                                                                2 + x*('A' - qsz), BASE_Y - n - y + 1, '
                                                                COLOR_BLACK, COLOR_WHITE, 5);
     showch(hout, BASE_X, BASE_Y + 40, '',
COLOR BLACK, COLOR WHITE, 1);
                                                                           /* 将第一个盘子从当前柱子下落到另一
                                                                根柱子 */
void move(int n, char qsz, char mbz, int h, int
                                                                for (luo = 0; luo < 18; luo++)
choice)//移动
                                                                      if (qsz != 'B') {
     HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
                                                                     showch (hout, BASE_X + 8 + (qsz - 'A') * 36
                                                                + 2 + x*('A' - qsz + 1) + (qsz - 'A'), BASE_Y - n - y + 1 + 1uo, '', 3, fg_color, 5);
     const int BASE_X = 10;
     const int BASE_Y = 28;
     const int fg_color = COLOR_HBLUE;
                                                                      Sleep (100);
     int y, x, luo, m;
                                                                      if (luo < 4)
     if (qsz == 'A')
                                                                         showch (hout, BASE_X + 8 + (qsz - 'A') *
```

装

订

```
36 + 2 + x*('A' - qsz + 1) + (qsz - 'A'), BASE_Y
                                                               qsz)); x++)
- n - y + 1 + 1uo, '', COLOR_BLACK,
COLOR_WHITE, 5);
                                                                if (qsz != 'B')
if (luo >= 4 && luo < 17)
                                                               showch (hout, BASE_X + 9 + (qsz - 'A') * 36 + 2 +
    showch (hout, BASE_X + 10 + 36 + 3, BASE_Y -
                                                               x*('A' - qsz + 1), BASE_Y - n - y + 1, ', 3 +
n - y + 1 + 1uo, ', COLOR_HYELLOW,
                                                                (h - 1), fg_{color}, 5 + 1 * (h - 1);
COLOR WHITE, 1);
                                                                if (choice == 8)
  showch (hout, BASE_X + 10 + 36 + 4, BASE_Y - n
                                                                 Sleep (100);
 y + 1 + 1uo, '', COLOR_BLACK, COLOR_WHITE,
                                                                if (x < 37 * abs((int)(mbz - qsz)))
                                                                   showch (hout, BASE_X + 9 + (qsz - 'A') * 36 +
2);
                                                               2 + x*('A' - qsz + 1), BASE Y - n - y + 1, '
 showch (hout, BASE X + 10 + 36 + 1, BASE Y - n
- y + 1 + 1uo, '', COLOR_BLACK, COLOR_WHITE,
                                                               COLOR_BLACK, COLOR_WHITE, 5 + 1 * (h - 1);
                                                                                if (qsz == 'B')
                                                                                {
                                                                                     showch(hout, BASE_X + 9 +
if (qsz = 'B')
                                                                (qsz - A') * 36 + 2 + x*(mbz - qsz), BASE_Y - n
                                                               -y + 1, ', 3 + (h - 1), fg_color, 5 + 2 * (h
showch(hout, BASE_X + 8 + (qsz - 'A') * 36 +
x*('A' - qsz) + 4, BASE_Y - n - y + 1 + 1uo, '
                                                               - 1));
', 3, fg_color, 5);
                                                                                      if (choice == 8)
Sleep(100);
                                                                                           Sleep (100);
if (1110 < 4)
                                                                                      if (x < 37)
  showch (hout, BASE_X + 8 + (qsz - 'A') * 36 +
                                                                                           showch(hout, BASE_X
x*('A' - qsz) + 4, BASE_Y - n - y + 1 + 1uo, '
                                                               + 9 + (qsz - A') * 36 + 2 + x*(mbz - qsz),
                                                               BASE_Y - n - y + 1, ', COLOR_BLACK,
 , COLOR_BLACK, COLOR_WHITE, 5);
if (luo >= 4 && luo < 17)
                                                               COLOR_WHITE, 5 + 2 * (h - 1);
  showch (hout, BASE_X + 8 + (qsz - 'A') * 36 + 2
                                                                           /* 将第一个盘子从当前柱子下落到另一
+ x*('A' - qsz) + 4, BASE_Y - n - y + 1 + 1uo,
 , COLOR_HYELLOW, COLOR_WHITE, 1);
                                                                根柱子 */
  showch(hout, BASE_X + 8 + (qsz - 'A') * 36 + 2
                                                                           if (qsz = 'A')
+ x*('A' - qsz) + 5, BASE_Y - n - y + 1 + 1uo, '
', COLOR_BLACK, COLOR_WHITE, 2);
                                                                                if (mbz == 'B')
 showch (hout, BASE_X + 8 + (qsz - 'A') * 36 + 2
+ x*('A' - qsz) + 2, BASE_Y - n - y + 1 + 1uo,
                                                                                      for (1uo = 0; 1uo < 18 -
', COLOR_BLACK, COLOR_WHITE, 2);
                                                               btop + 1; luo++)
                     }
                                                                                           showch(hout, BASE_X
                                                               + 8 + (qsz - 'A') * 36 + 1 + x*('A' - qsz + 1) +
                                                                (C' - mbz) - (h - 1), BASE_Y - n - y + 1 + 1uo,
     if (choice == 8 || choice == 9)
                                                                ', 3 + (h - 1), fg\_color, 5 + 2 * (h - 1);
                                                                                            if (choice == 8)
     for (y = m; y < 19 - n; y++)
                                                                                                 Sleep (100);
           {
                                                                                            if (1uo < 4)
                showch (hout, BASE_X + 9 + (qsz)
                                                                                                showch (hout,
                                                               BASE_X + 8 + (qsz - 'A') * 36 + 1 + x*('A' - qsz)
- 'A') * 36 + 2 - (h - 1), BASE_Y - y - 1, '
                                                               + 1) + ('C' - mbz) - (h - 1), BASE_Y - n - y + 1
3 + h - 1, fg_color, 5 + 2 * (h - 1);
                if (choice == 8)
                                                               + 1uo, '', COLOR_BLACK, COLOR_WHITE, 5 + 2 * (h)
                      Sleep (100);
                                                               - 1));
                if (y < 15 - n)
                                                                                            if (luo >= 4 && luo
                                                               < 18 - btop)
                      showch (hout, BASE X + 9 +
                                                                                            {
(qsz - A') * 36 + 4, BASE_Y - y - 1, '
                                                                                                 showch (hout,
COLOR_HYELLOW, COLOR_WHITE, 1);
                                                               BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
                                                                1) + 2 + ('C' - mbz), BASE_Y - n - y + 1 + luo,
                     showch (hout, BASE_X + 9 +
(qsz - A') * 36 + 2 - (h - 1), BASE_Y - y - 1,
                                                                ' ', COLOR_HYELLOW, COLOR_WHITE, 1);
  ', COLOR_BLACK, COLOR_WHITE, 2 + 1 * (h - 1);
                                                                                                 showch (hout,
                                                               BASE X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
                     showch (hout, BASE X + 9 +
                                                               1) + 3 + ('C' - mbz), BASE_Y - n - y + 1 + 1uo,
(qsz - A') * 36 + 5, BASE_Y - y - 1, '
COLOR BLACK, COLOR WHITE, 2 + 1 * (h - 1);
                                                                ', COLOR BLACK, COLOR WHITE, 2 + 1 * (h - 1);
                                                                                                 showch (hout,
                 if (y >= 15 - n\&\& y < 19 - n)
                                                               BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
                                                               1) + ('C' - mbz) - (h - 1), BASE_Y - n - y + 1 +
                     showch (hout, BASE_X + 9 +
(qsz - A') * 36 + 2 - (h - 1), BASE_Y - y - 1,
                                                                     ', COLOR_BLACK, COLOR_WHITE, 2 + 1 * (h -
  ', COLOR_BLACK, COLOR_WHITE, 5 + 2 * (h - 1));
                                                               1)):
           for (x = 0; x < 37 * abs((int)) (mbz - 
                                                                                     }
```

装

订

线

```
else
                      for (luo = 0; luo < 18 -
ctop + 1; 1uo++)
                            showch(hout, BASE_X
+ 8 + (qsz - 'A') * 36 + 1 + x*('A' - qsz + 1) +
                                                                                 }
('C' - mbz) - (h - 1), BASE_Y - n - y + 1 + 1uo,
' ', 3 + (h - 1), fg\_color, 5 + 2 * (h - 1);
                                                                                 {
                            if (choice = 8)
                                  Sleep (100);
                                                                btop + 1; luo++)
                            if (1uo < 4)
                                 showch (hout,
BASE_X + 8 + (qsz - 'A') * 36 + 1 + x*('A' - qsz)
+ 1) + ('C' - mbz) - (h - 1), BASE_Y - n - y + 1
+ 1uo, '', COLOR_BLACK, COLOR_WHITE, 5 + 2 * (h
                            if (luo >= 4 && luo
< 18 - ctop)
                                  showch (hout,
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
1) + 2 + (C' - mbz), BASE Y - n - y + 1 + 1uo,
                                                                + 1uo,
' ', COLOR HYELLOW, COLOR WHITE, 1);
                                                                - 1)):
                                  showch (hout,
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
                                                                < 18 - btop)
1) + 3 + ('C' - mbz), BASE_Y - n - y + 1 + 1uo,
, , COLOR_BLACK, COLOR_WHITE, 2 + 1 * (h - 1);
                                 showch (hout,
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
1) + ('C' - mbz) - (h - 1), BASE_Y - n - y + 1 +
1uo, '', COLOR_BLACK, COLOR_WHITE, 2 + 1 * (h -
1));
                 }
           if (qsz == 'C')
                 if (mbz == 'A')
                                                                (h - 1));
                      for (1uo = 0; 1uo < 18 -
atop + 1; luo++)
                                                                                 }
                           showch (hout, BASE_X
+ 8 + (q_{SZ} - 'A') * 36 + 5 + x*('A' - q_{SZ} + 1) -
(h - 1) + (A' - mbz), BASE_Y - n - y + 1 + 1uo,
                                                                if (mbz == 'A')
  ', 3 + (h - 1), fg\_color, 5 + 2 * (h - 1);
                                                                {
                            if (choice = 8)
                                 Sleep (100):
                            if (luo < 4)
                                 showch (hout,
BASE_X + 8 + (qsz - 'A') * 36 + 5 + x*('A' - qsz)
+1) - (h-1) + (A' - mbz), BASE Y - n - y + 1
+ 1uo, '', COLOR_BLACK, COLOR_WHITE, 5 + 2 * (h
- 1));
                            if (1uo >= 4 && 1uo
                                                                      if (luo < 4)
< 18 - atop)
                            {
                                 showch (hout,
BASE_X + 9 + (qsz - A') * 36 + x*(A' - qsz +
1) + 6 + ('A' - mbz), BASE_Y - n - y + 1 + 1uo,
 ', COLOR_HYELLOW, COLOR_WHITE, 1);
                                 showch (hout,
BASE_X + 9 + (q_{SZ} - 'A') * 36 + x*('A' - q_{SZ} +
1) + 7 + (A' - mbz), BASE_Y - n - y + 1 + 1uo,
```

', COLOR\_BLACK, COLOR\_WHITE, 2 + 1 \* (h - 1);

```
showch (hout,
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
1) + 4 - (h - 1) + ('A' - mbz), BASE_Y - n - y +
1 + 1uo, ', COLOR_BLACK, COLOR_WHITE, 2 + 1 *
                  else
                         for (1uo = 0; 1uo < 18 -
                              showch (hout, BASE X
+ 8 + (qsz - 'A') * 36 + 5 + x*('A' - qsz + 1) +
(A' - mbz) - (h - 1), BASE_Y - n - y + 1 + 1uo,
 ', 3 + (h - 1), fg\_color, 5 + 2 * (h - 1);
                               if (choice == 8)
                                     Sleep(100);
                               if (1uo < 4)
                                    showch (hout,
BASE X + 8 + (qsz - 'A') * 36 + 5 + x*('A' - qsz)
+ 1) + (A' - mbz) - (h - 1), BASE Y - n - y + 1
         ', COLOR_BLACK, COLOR_WHITE, 5 + 2 * (h
                               if (1uo >= 4 && 1uo
                                    showch (hout,
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
1) + 6 + ('A' - mbz), BASE_Y - n - y + 1 + 1uo,
 ', COLOR_HYELLOW, COLOR_WHITE, 1);
                                    showch (hout,
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz +
1) + 7 + ('A' - mbz), BASE_Y - n - y + 1 + 1uo,
' ', COLOR_BLACK, COLOR_WHITE, 2 + 1 * (h - 1));
BASE_X + 9 + (qsz - 'A') * 36 + x*('A' - qsz + 1) + 4 - (h - 1) + ('A' - mbz), BASE_Y - n - y + 1 + 1uo, '', COLOR_BLACK, COLOR_WHITE, 2 + 1 *
            if (qsz == 'B')
    for (luo = 0; luo < 18 - atop + 1; luo++)
      showch (hout, BASE X + 8 + (qsz - 'A') * 36
+ x*(mbz - qsz) + 4 - (mbz - 'A') - (h - 1),
BASE_Y - n - y + 1 + 1uo, ', 3 + (h - 1),
fg\_color, 5 + 2 * (h - 1));
      if (choice == 8)
            Sleep (100);
        showch (hout, BASE X + 8 + (qsz - 'A') *
36 + x*(mbz - qsz) + 4 - (mbz - 'A') - (h - 1),
BASE Y - n - y + 1 + 1uo, '', COLOR BLACK,
COLOR_WHITE, 5 + 2 * (h - 1);
      if (luo >= 4 && luo < 18 - atop)
       showch (hout, BASE_X + 8 + (qsz - 'A') *
36 + 2 + x*(mbz - qsz) + 4 - (mbz - 'A'), BASE_Y - n - y + 1 + 1uo, '', COLOR_HYELLOW,
COLOR WHITE, 1);
```

订

```
showch (hout, \ BASE_X + 8 + (qsz - 'A') * 36 \\ + 2 + x*(mbz - qsz) + 5 - (mbz - 'A'), \ BASE_Y - \\ n - y + 1 + luo, '', \ COLOR_BLACK, \ COLOR_WHITE,
                                                                               cout << setiosflags(ios::right) << setw(2)</pre>
                                                                         << "B: ";
2 + 1 * (h - 1));
                                                                               for (i = 0, j = 0; i < N; i++)
     showch (hout, BASE_X + 8 + (qsz - 'A') * 36
+ 2 + x*(mbz - qsz) + 2 - (h - 1) - (mbz - A'),
BASE_Y - n - y + 1 + 1uo, '', COLOR_BLACK,
                                                                                     if (b[i] == 0)
COLOR_WHITE, 2 + 1 * (h - 1));
                                                                                            for (j; j < N - i; j++)
                                                                                               cout << setw(2) << " ";
                   }
                                                                                     }
else
                                                                                     else
                                                                                            \texttt{cout} \; << \; \texttt{setw(1)} \; << \; \texttt{b[i]} \; << \; \texttt{"} \; \texttt{"};
    for (1uo = 0; 1uo < 18 - ctop + 1; 1uo++)
              showch (hout, BASE_X + 8 + (qsz -
                                                                               cout << setiosflags(ios::right) << setw(2)</pre>
'A') * 36 + x*(mbz - qsz) + 4 - (mbz - 'A') - (h - 1), BASE_Y - n - y + 1 + 1uo, '', 3 + (h -
                                                                         << "C: ";
                                                                               for (i = 0; i \langle N; i++)
1), fg_color, 5 + 2 * (h - 1);
            if (choice == 8)
                                                                                     if (c[i] == 0)
               Sleep(100);
                                                                                            for (j; j < N - i; j++)
             if (1uo < 4)
                                                                                              cout << setw(2) << " ";
            showch (hout, BASE_X + 8 + (qsz -
'A') * 36 + x*(mbz - qsz) + 4 - (mbz - 'A') - (h
                                                                                            break:
- 1), BASE_Y - n - y + 1 + 1uo, ',',
                                                                                     }
COLOR_BLACK, COLOR_WHITE, 5 + 2 * (h - 1));
                                                                                     else
           if (luo >= 4 && luo < 18 - ctop)
                                                                                            cout << setw(1) << c[i] << " ";
              showch (hout, BASE_X + 8 + (qsz -
                                                                               cout << end1;</pre>
'A') * 36 + 2 + x*(mbz - qsz) + 4 - (mbz - 'A'),
BASE_Y - n - y + 1 + 1uo, , COLOR_HYELLOW,
COLOR_WHITE, 1);
                                                                         void shuchushihua()
            showch (hout, BASE_X + 8 + (qsz -
'A') * 36 + 2 + x*(mbz - qsz) + 5 - (mbz - 'A'),
                                                                               const int BASE_X = 10;
BASE_Y - n - y + 1 + 1uo, , COLOR_BLACK,
                                                                               const int BASE_Y = 40;
COLOR_WHITE, 2 + 1 * (h - 1);
                                                                               int y, k;
showch (hout, BASE_X + 8 + (qsz - 'A') * 36 + 2 +
                                                                               HANDLE hout =
x*(mbz - qsz) + 2 - (h - 1) - (mbz - A),
BASE_Y - n - y + 1 + 1uo, ' ', COLOR_BLACK,
                                                                         GetStdHandle(STD_OUTPUT_HANDLE);
                                                                               gotoxy(hout, BASE_X, BASE_Y);
COLOR_WHITE, 2 + 1 * (h - 1);
                                                                               cout << "A";
                                                                               gotoxy (hout, BASE_X + 1 * 10, BASE Y);
                                                                               cout << "B";
                                                                               gotoxy(hout, BASE_X + 2 * 10, BASE_Y);
                                                                               cout << "C";
      showch(hout, BASE_X, BASE_Y + 40, '',
                                                                               gotoxy(hout, BASE_X - 10, BASE_Y - 1);
COLOR_BLACK, COLOR_WHITE, 1);
                                                                               cout <<
void hengchushihua()
                                                                         end1:
                                                                               for (k = 0; k < N; k++)
       const int BASE X = 10;
       const int BASE Y = 40;
                                                                                     gotoxy(hout, BASE_X, BASE_Y - k -
                                                                         2):
       int i, i:
       HANDLE hout =
GetStdHandle(STD OUTPUT HANDLE); //取标准输出设
                                                                                     if (k \ge 0 \&\& k \le atop - 1)
备对应的句柄
                                                                                           cout \ll a[k];
                                                                                     if (k \ge atop \&\& k \le 10)
       gotoxy(hout, BASE_X, BASE_Y + 2);
                                                                                           cout << " ";
      cout << "初始:";//输出初始行
cout << "A: ";
                                                                               for (k = 0; k < N; k++)
      for (i = 0, j = 0; i < N; i++)
                                                                                     gotoxy(hout, BASE_X + 10, BASE_Y - k
                                                                         - 2);
             if (a[i] = 0)
                   for (j; j < N - i; j++)
                                                                                     if (k >= 0 \&\& k <= btop - 1)
                        cout << setw(2) << " ";
                                                                                            cout \langle\langle b[k];
                                                                                      if (k \ge btop\&\& k \le 10)
                   break:
                                                                                            cout << " ";
                   cout << setw(1) << a[i] << " ";
                                                                               for (y = 19, k = 0; y > 9 \&\& k < N; y--,
```

装

订

```
k++)
                                                                 BASE_Y + 2);
                                                                                   if (choice == 3)
           gotoxy(hout, BASE_X + 20, BASE_Y - k
                                                                                        gotoxy(hout, 0, num - 1);
- 2);
                                                                                   go(from, to, h, num, choice);
                                                                                   if (choice \geq= 4)
           if (k \ge 0 \&\& k \le ctop - 1)
                                                                                         shu();
                 cout << c[k];
                                                                                   move(n, from, to, h, choice);
           if (k \ge \cot k \cdot k \le 10)
                                                                                   num++:
                 cout << " ";
                                                                             if (from == 'A'&&to == 'B')
}
void shu()//纵向输出
                                                                                   h = b[btop++] = a[--atop];
                                                                                   a[atop] = 0;
     const int BASE X = 10;
                                                                                   if (sleep == 0)
      const int BASE Y = 40;
                                                                                         gsy = _getch();
     HANDLE hout =
                                                                                         Sleep((6 - sleep) * 100);
GetStdHandle(STD_OUTPUT_HANDLE);
                                                                                   if (choice \geq= 4)
     int y, k;
      for (k = 0; k < N; k++)
                                                                                         gotoxy(hout, BASE_X,
                                                                 BASE_Y + 2);
      gotoxy(hout, BASE_X, BASE_Y - k - 2);
                                                                                   if (choice == 3)
           if (k >= 0 \&\& k <= atop - 1)
                                                                                         gotoxy(hout, 0, num - 1);
                cout \ll a[k];
                                                                                   go(from, to, h, num, choice);
           if (k \ge atop \&\& k \le 10)
                                                                                   if (choice >= 4)
                cout << " ";
                                                                                        shu();
                                                                                   move(n, from, to, h, choice);
     for (k = 0; k < N; k++)
                                                                                   num++;
     gotoxy(hout, BASE_X + 10, BASE_Y - k - 2)
                                                                             if (from == 'B'&&to == 'C')
           if (k \ge 0 \&\& k \le btop - 1)
                cout \ll b[k];
                                                                                   h = c[ctop++] = b[--btop];
                                                                                   b[btop] = 0;
           if (k \ge btop\&\& k \le 10)
                cout << " ";
                                                                                   if (sleep == 0)
                                                                                        gsy = _getch();
      for (y = 19, k = 0; y > 9 \&\& k < N; y--,
k++)
                                                                                         Sleep((6 - sleep) * 100);
                                                                                   if (choice \geq= 4)
      gotoxy(hout, BASE_X + 20, BASE_Y - k - 2);
                                                                                         gotoxy(hout, BASE_X,
           if (k \ge 0 \&\& k \le ctop - 1)
                                                                 BASE_Y + 2);
                                                                                   if (choice == 3)
                cout << c[k];
           if (k \ge \cot k \cdot k \le 10)
                                                                                         gotoxy(hout, 0, num - 1);
                 cout << " ";
                                                                                   go(from, to, h, num, choice);
                                                                                   if (choice \geq= 4)
                                                                                         shu();
int hanoidg(char from, char to, int sleep, int
                                                                                   move(n, from, to, h, choice);
n, int choice, int t)//内部数组
      static int num = 1;
                                                                             if (from == 'B'&&to == 'A')
      if (t == 0)
           num = 1;
                                                                                   h = a[atop++] = b[--btop];
      if (t == 1)
                                                                                   b\lceil btop \rceil = 0:
                                                                                   if (sleep == 0)
      {
           int h;//盘号
                                                                                        gsy = _getch();
           int gsy;
                                                                                   else
           const int BASE X = 10;
                                                                                         Sleep ((6 - sleep) * 100);
           const int BASE_Y = 40;
                                                                                   if (choice >= 4)
           HANDLE hout =
                                                                                         gotoxy(hout, BASE_X,
GetStdHandle(STD_OUTPUT_HANDLE);
                                                                 BASE_Y + 2);
           if (from == 'A'&&to == 'C')
                                                                                   if (choice == 3)
                                                                                         gotoxy(hout, 0, num - 1);
                 h = c[ctop++] = a[--atop];
                                                                                   go(from, to, h, num, choice);
                 a[atop] = 0;
                                                                                   if (choice \geq= 4)
                 if (sleep = 0)
                                                                                         shu();
                       gsy = getch();
                                                                                   move(n, from, to, h, choice);
                                                                                   num++:
                       Sleep((6 - sleep) * 100);
                                                                             if (from == 'C'&&to == 'A')
                 if (choice \geq= 4)
                       gotoxy(hout, BASE_X,
```

装

订

```
h = a[atop++] = c[--ctop];
                                                                                cout << "第" <<
                                                                     setiosflags(ios::right) << setw(4) << num << "
                  c[ctop] = 0;
                  if (sleep = 0)
                                                                     步" << "(" << setiosflags(ios::right) << setw(2)
                                                                    << h << "#: " << from << "-->" << to << ") " <<
                       gsy = getch();
                                                                    end1;
                        Sleep ((6 - sleep) * 100);
                  if (choice \geq= 4)
                                                                          if (choice \geq= 3)
                        gotoxy(hout, BASE_X,
BASE_Y + 2);
                                                                                cout << "第" <<
                  if (choice == 3)
                                                                    setiosflags(ios::right) << setw(4) << num << "
                       gotoxy(hout, 0, num - 1);
                                                                    步" << "(" << setiosflags(ios::right) << setw(2) << h << "#: " << from << "-->" << to << ") ";
                  go(from, to, h, num, choice);
                  if (choice \geq= 4)
                                                                                cout << "A: ";
                       shu();
                                                                                for (i = 0, j = 0; i < N; i++)
                  move(n, from, to, h, choice);
                                                                                      \text{if } (a[i] = 0) \\
           if (from == 'C'&&to == 'B')
                                                                                       {
                                                                                            for (j; j < N - i; j++)
                                                                                                  cout << setw(2) << "
                  h = b[btop++] = c[--ctop];
                  c[ctop] = 0;
                  if (sleep = 0)
                                                                                            break;
                       gsy = _getch();
                                                                                      }
                        Sleep ((6 - sleep) * 100);
                                                                                            cout << setw(1) << a[i]
                  if (choice \geq= 4)
                        gotoxy(hout, BASE_X,
BASE_Y + 2);
                                                                                cout << setiosflags(ios::right) <<</pre>
                                                                    setw(2) << "B: ";
                  if (choice == 3)
                                                                    for (i = 0, j = 0; i < N; i++)
                        gotoxy (hout, 0, num - 1);
                  go(from, to, h, num, choice);
                  if (choice \geq = 4)
                                                                      if (b[i] = 0)
                        shu();
                                                                         for (j; j < N - i; j++)
                  move(n, from, to, h, choice);
                                                                          cout << setw(2) << " ";
                  num++;
                                                                               break;
           }
                                                                           else
                                                                           cout << setw(1) << b[i] << " ";
      return 0;
void DG(int n, char one, char two, char three,
                                                              \texttt{cout} \, <\!<\, \texttt{setiosflags(ios::right)} \, <\!<\, \texttt{setw(2)} \, <\!<\, \text{\texttt{"C: ";}}
int sleep, int choice, int t)//递归函数
                                                                    for (i = 0; i < N; i++)
{
                                                                      if (c[i] = 0)
      char jl = three;
      if (n == 1)
          hanoidg(one, three, sleep, n,
                                                                           for (j; j < N - i; j++)
                                                                                cout << setw(2) << " ";
choice, t);
                                                                                            break;
  \mbox{DG}\,(\mbox{n} - 1, one, three, two, sleep, choice, t);
                                                                                      }
  hanoidg(one, three, sleep, n, choice, t);
                                                                                      cout << setw(1) << c[i] << " ";
  DG(n-1, two, one, three, sleep, choice, t);
                                                                                cout << end1;</pre>
void go(char from, char to, int h, int num, int
choice)//横向输出
                                                                    }
                                                                    void game (int n, char qsz, char mbz, int choice)
     HANDLE hout =
GetStdHandle(STD_OUTPUT_HANDLE);
                                                                           static int num = 1, h;
      int i, j;
                                                                          while (1)
      if (choice == 1)
                                                                                const int BASE X = 10;
            gotoxy(hout, 0, num - 1);
                                                                                const int BASE_Y = 28;
           cout << h << "#: " << from << "-->"
                                                                                HANDLE hout =
                                                                    GetStdHandle(STD_OUTPUT_HANDLE);
<< to << endl:
                                                                                gotoxy(hout, BASE_X, BASE_Y + 14);
      if (choice == 2)
                                                                                \mathtt{cout} \, <\!< \, {''}
                                                                    " << endl;
            gotoxy(hout, 0, num - 1);
                                                                                gotoxy(hout, BASE_X, BASE_Y + 14);
```

装

订

```
get_int("请输入移动的柱号(命令形
                                                                                  go(qsz, mbz, h, num, choice);
式: AC=A顶端的盘子移动到C,Q=退出):", 'a', 'c',
                                                                                  shu();
&qsz, &mbz);
                                                                                  num++;
           if (qsz == 'q' \mid | qsz == 'Q')
                                                                            if (qsz == 'A'&&mbz == 'B')
                 gotoxy(hout, BASE_X + 4, BASE_Y
+ 15);
                 cout << "游戏中止!!!! " <<
                                                                                  h = b[btop++] = a[--atop];
end1;
                                                                                  a[atop] = 0;
                 break;
                                                                                  gotoxy(hout, BASE_X, BASE_Y +
                                                                 16);
                                                                                  go(qsz, mbz, h, num, choice);
           int m, j, k;
           if (mbz == 'A')
                                                                                  shu():
                                                                                  num++;
                 m = atop;
                 k = a[atop - 1];
                                                                            if (qsz == 'B' \&\&mbz == 'C')
           if (mbz == 'B')
                                                                                  h = c[ctop++] = b[--btop];
                                                                                  b[btop] = 0;
                 m = btop;
                                                                                  gotoxy(hout, BASE_X, BASE_Y +
                 k = b[btop - 1];
                                                                 16);
                                                                                  go(qsz, mbz, h, num, choice);
           if (mbz == 'C')
                                                                                  shu():
                                                                                  num++;
                 m = ctop:
                                                                            if (qsz == 'B' \&\&mbz == 'A')
                 k = c[ctop - 1];
           if (qsz == 'A')
                                                                                  h = a[atop++] = b[--btop];
                                                                                  b[btop] = 0;
                 j = a[atop - 1];
                                                                                  gotoxy(hout, BASE_X, BASE_Y +
                                                                 16);
           if (qsz == 'B')
                                                                                  go(qsz, mbz, h, num, choice);
                                                                                  shu();
                 j = b[btop - 1];
                                                                                  num++;
           if (qsz == 'C')
                                                                            if (qsz == 'C' \&\&mbz == 'A')
                 j = c[ctop - 1];
                                                                                  h = a[atop++] = c[--ctop];
           if (j == 0)
                                                                                  c[ctop] = 0;
                                                                                  gotoxy(hout, BASE_X, BASE_Y +
                 gotoxy(hout, BASE_X + 4, BASE_Y
                                                                 16);
+ 15);
                                                                                  go(qsz, mbz, h, num, choice);
                 cout << "源柱为空!" << endl;
                                                                                  shu();
                 Sleep(300);
                                                                                  \operatorname{num}^{++};
                 gotoxy(hout, BASE_X + 4, BASE_Y
                                                                            if (qsz == 'C' \&\&mbz == 'B')
+ 15);
                 \mathrm{cout} \, << \, {''}
                                    '' << end1;
                                                                                  h = b[btop^{++}] = c[--ctop];
                 continue;
                                                                                  c[ctop] = 0;
           if (k != 0)
                                                                                  gotoxy(hout, BASE X, BASE Y +
                 if (j > k)
                                                                 16);
{
                                                                                  go(qsz, mbz, h, num, choice);
     gotoxy(hout, BASE_X + 4, BASE_Y + 15);
                                                                                  shu();
     cout << "大盘压小盘, 非法移动! " << end1;
                                                                                  num++;
     Sleep (300);
     gotoxy(hout, BASE_X + 4, BASE_Y + 15);
                      cout <<
                                                                            move(n, qsz, mbz, h, choice);
" << end1;
                                                                            if (m + 1 == n)
                      continue:
                                                                                  gotoxy (hout, BASE X + 4, BASE Y
           if (qsz == 'A'&&mbz == 'C')
                                                                 + 15):
                                                                                  cout << "游戏结束!!!!" << end1;
                 h = c[ctop++] = a[--atop];
                                                                                  break;
                 a[atop] = 0;
                 gotoxy(hout, BASE_X, BASE_Y +
                                                                      }
16);
```

订