[(13条消息) C语言调整控制台颜色、大小、标题、文字位置、窗口位置、按钮\_Ricochesta的博客-CSDN博客\_c语言调节控制台字体大小](https://blog.csdn.net/cjz2005/article/details/104358000)白子：○ 黑子：●

黑先、白后

棋盘用20\*20二维数组，相应位置无子记为0，白子记为1，黑子记为2。

1.棋盘界面：遍历数组在相应位置输出对应符号，注意光标需要移动到(i，2\*j)，即gotoxy（2\*j,i），因为 程序框长：宽=2：1，否则会使符号变形。

2.下棋思路：用函数获取鼠标点击时的坐标，利用gotoxy移动光标至特定位置输出特定字符。

a.获取到鼠标点击位置在窗口区的位置（p.x,p.y）

[(13条消息) 科技系列 - 1 ：判断键盘上的某个键按下 ：KEY\_DOWN\_古呱的博客-CSDN博客\_windows监测键盘按下](https://blog.csdn.net/Hox_5/article/details/118901381)

b.将其换算成棋盘坐标

坐标原点位置为(a,b),字体一般长为m,宽为n且n:m=2:1,而棋盘上一格的边长为t=2n=4m,判定区域为r，经测试可得，a=b=t/4。

当a+tk-r<p <a+tk+r,可将其换算为棋盘坐标k(k为自然数)，可解得(p -r-a)/ t<k<(p+r-a)/ t,若该不等式中间只有一个唯一的整数，则2r/t<1,即r<t/2,结合实际需求,r=t/4较为合适。

将上述值代入不等式中可得(2p -t)/2t <k< p/t。则可能的k=[p/t]。再验证k>(2p -t)/2t,即k==[(2p -t)/2t]+1。

p.x的换算：将t=4m带入得，k=[p.x/4m] = [(p.x-2m)/4m] + 1

p.y的换算：将t=2n带入得，k= [p.y/2n] =[(p.y-n)/2n] + 1

需要额外注意的是，当k=0时，不管上限还是下限计算机计算后都为0，所以要特殊处理，即0<p.x <2m,0<p.y<n

c.将棋盘坐标换算为窗口坐标后在对应位置打印出来

gotoxy(4\*x,2\*y);

3.对弈思路：

首先进行先后手判定：若棋盘为空则直接正中落子。

对弈原则：

1. 一般情况下，进攻分数大于防守分数。
2. 其他分数按照：连五>活四>冲四>活三>冲三>活二>冲二,
3. 特殊加分机制：

必输（必胜）棋局：

一级(一步必死)：下一步形成连五

二级（两步必死）：下一步形成两个即以上冲四或一个以上活四

三级（三步必死）：下一步形成两个及以上活三

若必输棋局和必胜棋局同时存在，则判断是否处于同级别的棋局，若己方级别高于或等于对方级别，则计算机因为有先手优势，优先进攻；反之，己方级别低于对方，优先防守

因此，需要确定一个合理的分数。

设己方连五L51，对方连五L52, 己方活四H41，对方活四H42, 己方冲四C41，对方冲四C42, 己方活三H31，对方活三H32, 己方冲三C31，对方冲三C32, 己方活二H21，对方活二H22,己方冲二C21，对方冲二C22

由于连五优先等级最高，故L51>8L52,即即使对方某个位置八个方向上都出现连五，但由于先手优势，所以自己只要成功连五，那就必胜，

同理，按照上面的分析L51>8\*L52,L52>8\*H41,

H41>8\*H42 ,H32>C31>C32>H21>H22>C21>C22

由上述不等式可基本确定各个分数之间的关系

但注意某些棋局只有存在两个或以上才具有威胁，平常情况下并不危险。比如C41，假如按照上面的不等式计算，C41>H42，但实际上，H42>>C41,因此不能这样简单的判断，需要计算次数后再下结论

连五和活四是本身就具有很大威胁，而冲四和活三只有当个数多了才具有威胁，所以需要计算次数额外加分，加分规则：H41>2C41,2C41>8\*H42,H42>2C42,2C42>2H31,2H31>2H32,2H32>8\*C41

当不同级别的棋局同时存在于同一个位置，优先选择级别更高的棋局进行加分，忽略级别较低的棋局

4.胜负判定：

方法一：在落子点横、竖、左斜、右斜四条直线上左右各5个位置依次进行检测判断是否有同色棋子，从而判断是否胜利

可能运用到的函数：

移动光标函数：

void gotoxy(int x,int y) {

COORD pos; //定义光标位置的结构体变量

pos.X=x;

pos.Y=y;

HANDLE hout = GetStdHandle(STD\_OUTPUT\_HANDLE);//获取控制台句柄

SetConsoleCursorPosition(hout,pos); //设置光标位置

}

隐藏光标函数：

void HideCursor() {

CONSOLE\_CURSOR\_INFO curInfo; //定义光标信息的结构体变量

curInfo.dwSize = 1; //如果没赋值的话，光标隐藏无效

curInfo.bVisible = FALSE; //将光标设置为不可见

HANDLE handle = GetStdHandle(STD\_OUTPUT\_HANDLE); //获取控制台句柄

SetConsoleCursorInfo(handle, &curInfo); //设置光标信息

} 异色 同色·

活四 +oooo+ 1000000 1000000

冲四 oooo+ ooo+o oo+oo 100000 50000

活三 +ooo+ +o+oo+ 15000 10000

眠三 ++ooo +o+oo o++oo o+o+o 5000 1000

活二 ++oo++ +o+o+ o++o 800 300

眠二 +++oo ++o+o +o++o o+++o 100 35

一个 15 7

0个 0

棋局识别：五子棋是“零和游戏”，对于对方来说最优的位置即为我方应封堵的位置

利用串的模式匹配，针对需特定评分的序列进行匹配，剩下的序列统一一个分数

printf(“┏┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┓\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┣╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋╋┫\n”);

printf(“┗┻┻┻┻┻┻┻┻┻┻┻┻┻┻┻┻┻┻┛\n”);

┏ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┳ ━ ┓

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ●━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┣ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ╋ ━ ┫

┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃ ┃

┗ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┻ ━ ┛