### 黑白棋算法

黑白棋的游戏规则很简单：棋盘共8\*8=64个格子，棋子分黑白两棋，每次白棋先走，若两粒白棋子将若干黑棋子无空隙的夹在中间则中间夹的黑棋子全部变成白棋子。黑棋亦然。直到棋盘无法再放下任何棋子，此时哪种棋子的个数多，哪种棋子便获胜。

C++代码：

#include<iostream>

#include<cstdio>

using namespace std;

int board[8][8],saveboard[60][8][8];

int cx,cy,col,pass,empty,black,white;

void init(){ //initialization

memset(board,-1,sizeof(board));

board[3][3]=0;

board[3][4]=1;

board[4][4]=0;

board[4][3]=1;

col=0;

pass=0;

empty=60;

black=2;

white=2;

}

int input(){

char s[1000]="";

scanf("%s",&s);

if(s[0]>='a' && s[0]<='h')

cy=s[0]-'a';

else if(s[0]>='A' && s[0]<='H')

cy=s[0]-'A';

else return 0;

if(s[1]>='1' && s[1]<='8'){

cx=s[1]-'1';

return 1;

}

return 0;

}

int judge(int x,int y){

int i,j,temp;

temp=(col+1)%2;

//left && up

if(board[x-1][y-1]==temp){

for(i=x-1,j=y-1; i>=0 && j>=0; i--,j--){

if(board[i][j]<0) break;

if(col==board[i][j]) return 1;

}

}

//up

if(board[x-1][y]==temp){

for(i=x-1; i>=0; i--){

if(board[i][y]<0) break;

if(col==board[i][y]) return 1;

}

}

//right && up

if(board[x-1][y+1]==temp){

for(i=x-1,j=y+1; i>=0 && j<8; i--,j++){

if(board[i][j]<0) break;

if(col==board[i][j]) return 1;

}

}

//right

if(board[x][y+1]==temp){

for(j=y+1; j<8; j++){

if(board[x][j]<0) break;

if(col==board[x][j]) return 1;

}

}

//right && down

if(board[x+1][y+1]==temp){

for(i=x+1,j=y+1; i<8 && j<8; i++,j++){

if(board[i][j]<0) break;

if(col==board[i][j]) return 1;

}

}

//down

if(board[x+1][y]==temp){

for(i=x+1; i<8; i++){

if(board[i][y]<0) break;

if(col==board[i][y]) return 1;

}

}

//left && down

if(board[x+1][y-1]==temp){

for(i=x+1,j=y-1; i<8 && j>=0; i++,j--){

if(board[i][j]<0) break;

if(col==board[i][j]) return 1;

}

}

//left

if(board[x][y-1]==temp){

for(j=y-1; j>=0; j--){

if(board[x][j]<0) break;

if(col==board[x][j]) return 1;

}

}

return 0;

}

void move(int x,int y){

int i,j,temp,count;

temp=(col+1)%2;

count=0;

//left && up

if(board[x-1][y-1]==temp){

for(i=x-1,j=y-1; i>=0 && j>=0; i--,j--){

if(board[i][j]<0) break;

if(col==board[i][j]){

while(i!=x){

board[++i][++j]=col;

count++;

}

count--;

break;

}

}

}

//up

if(board[x-1][y]==temp){

for(i=x-1; i>=0; i--){

if(board[i][y]<0) break;

if(col==board[i][y]){

while(i!=x){

board[++i][y]=col;

count++;

}

count--;

break;

}

}

}

//right && up

if(board[x-1][y+1]==temp){

for(i=x-1,j=y+1; i>=0 && j<8; i--,j++){

if(board[i][j]<0) break;

if(col==board[i][j]){

while(i!=x){

board[++i][--j]=col;

count++;

}

count--;

break;

}

}

}

//right

if(board[x][y+1]==temp){

for(j=y+1; j<8; j++){

if(board[x][j]<0) break;

if(col==board[x][j]){

while(j!=y){

board[x][--j]=col;

count++;

}

count--;

break;

}

}

}

//right && down

if(board[x+1][y+1]==temp){

for(i=x+1,j=y+1; i<8 && j<8; i++,j++){

if(board[i][j]<0) break;

if(col==board[i][j]){

while(i!=x){

board[--i][--j]=col;

count++;

}

count--;

break;

}

}

}

//down

if(board[x+1][y]==temp){

for(i=x+1; i<8; i++){

if(board[i][y]<0) break;

if(col==board[i][y]){

while(i!=x){

board[--i][y]=col;

count++;

}

count--;

break;

}

}

}

//left && down

if(board[x+1][y-1]==temp){

for(i=x+1,j=y-1; i<8 && j>=0; i++,j--){

if(board[i][j]<0) break;

if(col==board[i][j]){

while(i!=x){

board[--i][++j]=col;

count++;

}

count--;

break;

}

}

}

//left

if(board[x][y-1]==temp){

for(j=y-1; j>=0; j--){

if(board[x][j]<0) break;

if(col==board[x][j]){

while(j!=y){

board[x][++j]=col;

count++;

}

count--;

break;

}

}

}

board[x][y]=col;

if(col){

black+=count;

white-=count;

black++;

}

else{

black-=count;

white+=count;

white++;

}

empty--;

}

void output(){

char c;

printf(" ");

for(int i=0; i<8; i++){

c='A'+i;

printf("%2c",c);

}

printf("\n");

for( i=0; i<8; i++){

printf("%d",i+1);

for(int j=0; j<8; j++){

if(board[i][j]==-1)

c=' ';

else if(board[i][j]==0)

c='O';

else

c='X';

printf("%2c",c);

}

printf("\n");

}

printf("Black:%3d White:%3d\n",black,white);

}

int passjudge(){

int f=0;

for(int i=0; i<8; i++)

for(int j=0; j<8; j++)

if(board[i][j]<0)

f+=judge(i,j);

return f;

}

void startprint(){

printf("1、New game\n2、setboard\n0、Exit\n");

}

void pvp(){

while(empty && pass<2){

//black or white

col++;

col%=2;

output();

//input

if(!input()){

if(!passjudge()){

printf("Pass!\n");

pass++;

}

else {

col++;

printf("No pass!\nPlease input right stone!\n");

}

continue;

}

if(judge(cx,cy)){

move(cx,cy);

pass=0;

}

else{

col++;

printf("Miss stone\n");

}

}

output();

if(black>white)

printf("Black Win!\n");

else if(black<white)

printf("White Win!\n");

else

printf("Draw Game!\n");

startprint();

}

void setboard(){

char c;

for(int i=0; i<8; i++)

for(int j=0; scanf("%c",&c) && c!='\n'; j++){

if(j>=8) continue;

if(c=='o' && c=='O')

board[i][j]=0;

else if(c=='x' && c=='X')

board[i][j]=1;

}

printf("White start or Black start?\n(W/B)");

scanf("%c",&c);

if(c=='w' || c=='W')

col=1;

if(c=='b' || c=='B')

col=0;

}

int main(int argc, char\* argv[]){

int n;

startprint();

while(scanf("%d",&n) && n){

init();

if(n==1)

pvp();

if(n==2){

setboard();

pvp();

}

}

return 0;

}

输出示例：

