#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//用宏来定义数组长度，方便将来对程序进行修改与扩展。

#define ROW 3

#define COL 3

//1.使用一个二维数组来定义一个棋盘。

char g\_chess\_board[ROW][COL];

void Init()

{

//TODO 对棋盘进行初始化

//光标放在字母上，用shift + alt + r 修改变量名。

for (int row = 0; row < ROW; ++row)

{

for (int col = 0; col < COL; ++col)

{

g\_chess\_board[row][col] = ' ';

}

}

}

void print()

{

//TODO 打印棋盘

for (int row = 0; row < ROW; ++row)

{

printf("| %c | %c | %c |\n",g\_chess\_board[row][0],

g\_chess\_board[row][1],g\_chess\_board[row][2]);

if (row != ROW)

{

//避免最后一行还输出一行隔断。

printf("|---|---|---|\n");

}

}

}

void PlayerMove()

{

while (1)

{

//TODO 玩家落子

//（1）提示玩家输入一个坐标。

printf("请玩家落子（输入坐标）\n");

//（2）读取玩家输入。

int row = -1;

int col = -1;

scanf("%d%d", &row, &col);

//（3）对玩家的输入进行校验。

//a.校验坐标范围是否合理

//b.校验玩家输入的坐标对应位置是否有棋子

if (row < 0 || row >= ROW || col < 0 || col >= COL)

{

printf("输入的坐标非法，请重新输入\n");

continue;

}

if (g\_chess\_board[row][col] != ' ')

{

printf("当前位置已经有子了，请重新输入\n");

continue;

}

//（4）如果玩家输入非法，就循环提示用户重新输入

//（5）如果玩家输入合法，就把棋盘上对应的位置设为‘x’。

g\_chess\_board[row][col] = 'x';

break;

}

}

//满了返回1，没满返回0

int IsFull()

{

for (int row = 0; row < ROW; row++)

{

for (int col = 0; col < COL; col++)

{

if (g\_chess\_board[row][col] == ' ')

{

return 0;

}

}

}

return 1;

}

//使用x表示玩家，使用o表示电脑。

//约定CheckWinner函数，返回x表示玩家胜利，返回o表示电脑胜利，

//返回q表示和棋，返回 空格 表示胜负未分。

char CheckWinner()

{

//TODO　输赢判定

//检测所有的行是不是三子连在一起

for (int row = 0; row < ROW; row++)

{

if (g\_chess\_board[row][0] == g\_chess\_board[row][1] &&

g\_chess\_board[row][0] == g\_chess\_board[row][2] &&

g\_chess\_board[row][0] != ' ')

{

return g\_chess\_board[row][0];

}

}

//再检测所有的列是不是连在一起

for (int col = 0; col < COL; col++)

{

if (g\_chess\_board[0][col] == g\_chess\_board[1][col] &&

g\_chess\_board[0][col] == g\_chess\_board[2][col] &&

g\_chess\_board[0][col] != ' ')

{

return g\_chess\_board[0][col];

}

}

//检测对角线

if (g\_chess\_board[0][0] == g\_chess\_board[1][1] &&

g\_chess\_board[0][0] == g\_chess\_board[2][2] &&

g\_chess\_board[0][0] != ' ')

{

return g\_chess\_board[0][0];

}

if (g\_chess\_board[2][0] == g\_chess\_board[1][1] &&

g\_chess\_board[2][0] == g\_chess\_board[0][2] &&

g\_chess\_board[2][0] != ' ')

{

return g\_chess\_board[2][0];

}

//检测是否和棋

if (IsFull())//使用IsFull函数来判断棋盘是不是满了

{

return 'q';

}

return ' ';

}

void ComputerMove()

{

//TODO 电脑落子

printf("请电脑落子\n");

int row = 0;

int col = 0;

while (1)

{

row = rand() % ROW;//把产生的随机数控制在[0,2]中

col = rand() % COL;//把产生的随机数控制在[0,2]中

if (g\_chess\_board[row][col] == ' ')

{

g\_chess\_board[row][col] = 'o';

break;

}

}

}

int main()

{

char winner = ' ';

//2.把棋盘全部初始化为空格。

Init();

while (1)

{

//3.打印棋盘。

print();

//4.玩家落子，并进行输赢判定。

PlayerMove();

winner = CheckWinner();

if (winner != ' ')

{

//游戏结束

break;

}

//5.电脑落子，并进行输赢判定。

ComputerMove();

winner = CheckWinner();

if (winner != ' ')

{

//游戏结束

break;

}

}

if (winner == 'x')

{

printf("玩家获胜！\n");

}

else if (winner == 'o')

{

printf("电脑获胜！\n");

}

else if (winner == 'q')

{

printf("和棋！\n");

}

system("pause");

return 0;

}