#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

#define PLAYER\_X 1

#define PLAYER\_O 2

#define EMPTY 0

int T[24]; // Représentation du plateau

// Fonction pour changer la couleur du texte et du fond

void color(int couleurDuTEXTE, int couleurDeFond) {

HANDLE H = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(H, couleurDeFond \* 16 + couleurDuTEXTE);

}

// Fonction pour afficher la grille avec couleur

void co(int i) {

if (T[i] == EMPTY) color(15, 0);

else if (T[i] == PLAYER\_X) color(4, 4);

else if (T[i] == PLAYER\_O) color(2, 2);

}

// Fonction pour calculer le score d'un joueur

int score(int joueur) {

int J = 0;

for (int i = 0; i < 24; i++) {

if (T[i] == joueur) J++;

}

return J;

}

// Fonction pour vérifier si un joueur est bloqué

int isBlocked(int joueur) {

for (int i = 0; i < 24; i++) {

if (T[i] == joueur) {

// Vérifiez si le joueur peut encore effectuer un mouvement

if ((i == 0 && T[1] == EMPTY) || (i == 1 && (T[0] == EMPTY || T[2] == EMPTY || T[4] == EMPTY)) ||

(i == 2 && T[1] == EMPTY) || (i == 3 && T[4] == EMPTY) || (i == 4 && (T[1] == EMPTY || T[3] == EMPTY || T[5] == EMPTY)) ||

(i == 5 && T[4] == EMPTY) || (i == 6 && T[3] == EMPTY) || (i == 7 && T[8] == EMPTY) || (i == 8 && (T[7] == EMPTY || T[9] == EMPTY || T[12] == EMPTY)) ||

(i == 9 && T[8] == EMPTY) || (i == 10 && T[13] == EMPTY) || (i == 11 && T[12] == EMPTY) || (i == 12 && (T[8] == EMPTY || T[11] == EMPTY || T[13] == EMPTY || T[16] == EMPTY)) ||

(i == 13 && T[10] == EMPTY) || (i == 14 && T[15] == EMPTY) || (i == 15 && (T[14] == EMPTY || T[16] == EMPTY || T[18] == EMPTY)) ||

(i == 16 && T[12] == EMPTY) || (i == 17 && T[18] == EMPTY) || (i == 18 && (T[15] == EMPTY || T[17] == EMPTY || T[19] == EMPTY)) ||

(i == 19 && T[18] == EMPTY) || (i == 20 && T[21] == EMPTY) || (i == 21 && (T[20] == EMPTY || T[22] == EMPTY || T[23] == EMPTY)) ||

(i == 22 && T[21] == EMPTY) || (i == 23 && T[22] == EMPTY)) {

return 0; // Le joueur peut encore jouer

}

}

}

return 1; // Le joueur est bloqué

}

// Fonction pour afficher la grille colorée

void grille() {

printf("\n\n\n");

printf("\t\t\t ");

co(0);

printf("|00|");

color(15, 0);

printf("=======================");

co(1);

printf("|01|");

color(15, 0);

printf("=======================");

co(2);

printf("|02|\n");

color(15, 0);

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

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

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

printf("\t\t\t | ");

co(3);

printf("|03|");

color(15, 0);

printf("============");

co(4);

printf("|04|");

color(15, 0);

printf("============");

co(5);

printf("|05|");

color(15, 0);

printf(" |\n");

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

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

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

printf("\t\t\t | | ");

co(6);

printf("|06|");

color(15, 0);

printf("======");

co(7);

printf("|07|");

color(15, 0);

printf("=====");

co(8);

printf("|08|");

color(15, 0);

printf(" | |\n");

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

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

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

printf("\t\t\t ");

co(9);

printf("|09|");

color(15, 0);

printf("=======");

co(10);

printf("|10|");

color(15, 0);

printf("==");

co(11);

printf("|11|");

color(15, 0);

printf(" ");

co(12);

printf("|12|");

color(15, 0);

printf("==");

co(13);

printf("|13|");

color(15, 0);

printf("========");

co(14);

printf("|14|");

printf("\n");

color(15, 0);

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

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

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

color(15, 0);

printf("\t\t\t | | ");

co(15);

printf("|15|");

color(15, 0);

printf("======");

co(16);

printf("|16|");

color(15, 0);

printf("=====");

co(17);

printf("|17|");

color(15, 0);

printf(" | |\n");

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

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

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

color(15, 0);

printf("\t\t\t | ");

co(18);

printf("|18|");

color(15, 0);

printf("============");

co(19);

printf("|19|");

color(15, 0);

printf("============");

co(20);

printf("|20|");

color(15, 0);

printf(" |\n");

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

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

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

printf("\t\t\t ");

co(21);

printf("|21|");

color(15, 0);

printf("=======================");

co(22);

printf("|22|");

color(15, 0);

printf("=======================");

co(23);

printf("|23|\n");

color(15, 0);

printf("\t\t\t score 1 :%d score 2 :%d \n\n\n", score(PLAYER\_X), score(PLAYER\_O));

color(15, 0);

}

void MOULIN(int i, int j, int k) {

if ((T[i] == PLAYER\_X && T[j] == PLAYER\_X && T[k] == PLAYER\_X) || (T[i] == PLAYER\_O && T[j] == PLAYER\_O && T[k] == PLAYER\_O)) {

printf("\n\n########## Player %d Wins ##############\n", T[i]);

exit(0);

}

}

int minimax(int depth, int maximizingPlayer) {

if (depth == 0 || score(PLAYER\_X) == 3 || score(PLAYER\_O) == 3 || (isBlocked(PLAYER\_X) && isBlocked(PLAYER\_O))) {

if (maximizingPlayer) return score(PLAYER\_O) - score(PLAYER\_X);

else return score(PLAYER\_X) - score(PLAYER\_O);

}

if (maximizingPlayer) {

int maxEval = -1000;

for (int i = 0; i < 24; i++) {

if (T[i] == EMPTY) {

T[i] = PLAYER\_O;

int eval = minimax(depth - 1, 0);

T[i] = EMPTY;

if (eval > maxEval) maxEval = eval;

}

}

return maxEval;

} else {

int minEval = 1000;

for (int i = 0; i < 24; i++) {

if (T[i] == EMPTY) {

T[i] = PLAYER\_X;

int eval = minimax(depth - 1, 1);

T[i] = EMPTY;

if (eval < minEval) minEval = eval;

}

}

return minEval;

}

}

void computerMove() {

int bestMove = -1;

int bestVal = -1000;

for (int i = 0; i < 24; i++) {

if (T[i] == EMPTY) {

T[i] = PLAYER\_O;

int moveVal = minimax(3, 0);

T[i] = EMPTY;

if (moveVal > bestVal) {

bestMove = i;

bestVal = moveVal;

}

}

}

T[bestMove] = PLAYER\_O;

grille();

MOULIN(0, 1, 2);

MOULIN(0, 3, 6);

MOULIN(0, 9, 15);

MOULIN(1, 4, 7);

MOULIN(3, 4, 5);

MOULIN(2, 5, 8);

MOULIN(6, 7, 8);

MOULIN(9, 10, 11);

MOULIN(12, 13, 14);

MOULIN(15, 16, 17);

MOULIN(18, 19, 20);

MOULIN(21, 22, 23);

MOULIN(10, 13, 16);

MOULIN(11, 14, 17);

MOULIN(12, 13, 14);

MOULIN(18, 19, 20);

MOULIN(19, 20, 21);

MOULIN(20, 21, 22);

if (score(PLAYER\_X) + score(PLAYER\_O) == 24) {

printf("\n\n########## Draw ##############\n");

exit(0);

}

}

void play(int i, int joueur) {

T[i] = joueur;

grille();

MOULIN(0, 1, 2);

MOULIN(0, 3, 6);

MOULIN(0, 9, 15);

MOULIN(1, 4, 7);

MOULIN(3, 4, 5);

MOULIN(2, 5, 8);

MOULIN(6, 7, 8);

MOULIN(9, 10, 11);

MOULIN(12, 13, 14);

MOULIN(15, 16, 17);

MOULIN(18, 19, 20);

MOULIN(21, 22, 23);

MOULIN(10, 13, 16);

MOULIN(11, 14, 17);

MOULIN(12, 13, 14);

MOULIN(18, 19, 20);

MOULIN(19, 20, 21);

MOULIN(20, 21, 22);

if (score(PLAYER\_X) + score(PLAYER\_O) == 24) {

printf("\n\n########## Draw ##############\n");

exit(0);

}

if (joueur == PLAYER\_X)

computerMove();

}

int main() {

}