איקס עיגול מול המחשב קוד

#include<stdio.h>

//מדפיס מיקומים

void location\_x\_o() {

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

printf("[%d]", 3 \* i + j);

}

printf("\n");

}

}

//מדפיס\_X\_O

void print(char mat[3][3]) {

printf("\n");

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

{

if (mat[i][j] == 'o' || mat[i][j] == 'x')

printf("[%c]", mat[i][j]);

else

printf("[ ]");

}

}

printf("\n");

}

}

//בודק אם יש מנצח ומי

int winner\_test(char mat[3][3]) {

int column = 0, row = 0, main\_diagonal = 0, secondary\_diagonal = 0;

int columnx = 0, rowx = 0, main\_diagonalx = 0, secondary\_diagonalx = 0;

int i, j, counter=0;

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

if (mat[i][j] == 'x' || mat[i][j] == 'o')//כללי

counter++;

if (mat[i][j] == 'o')

row++;

else if (mat[i][j] == 'x')

rowx++;

if (mat[j][i] == 'o')

column++;

else if (mat[j][i] == 'x')

columnx++;

if (mat[j][j] == 'o')

main\_diagonal++;

else if (mat[j][j] == 'x')

main\_diagonalx++;

if (mat[j][2-j] == 'o')

secondary\_diagonal++;

else if (mat[j][2-j] == 'x')

secondary\_diagonalx++;

}

if (column == 3 || row == 3 || main\_diagonal == 3 || secondary\_diagonal == 3)

return 1;

if (columnx == 3 || rowx == 3 || main\_diagonalx == 3 || secondary\_diagonalx == 3)

return 2;

column = 0; row = 0; main\_diagonal = 0; secondary\_diagonal = 0;

columnx = 0; rowx = 0; main\_diagonalx = 0; secondary\_diagonalx = 0;

}

if (counter >=9) {

return 3;

}

return 0;

}

//קלט איק עיגול מיקום והשמה

void scan\_x\_o(char mat[3][3], char xo) {

int num;

printf("plaese anter location ; ");

scanf\_s("%d", &num);

while (mat[0][num] != 0 || num < 0 || num>8)

{

printf("plaese anter a blank location ; ");

scanf\_s("%d", &num);

}

mat[0][num] = xo;

}

//מי מתחיל

int strat\_x\_o() {

int num;

do

{

printf("plaese enter 0 to start 1 to be second : ");

scanf\_s("%d", &num);

} while (num<0||num>1);

return num;

}

//חלק 1

//אם לכל היותר משבצת אחת מסומנת על הלוח וגם המשבצת האמצעית פנויה – לסמן את המשבצת האמצעית

int algorithm\_one(char mat[3][3],char me,char he,int counter){

if(counter>=1&& mat[1][1]==0){

mat[1][1] = me;

return 1;

}

return 0;

}

//חלק 2

//אם בשורה או באלכסון מסומנות שתי משבצות שלי – לסמן את המשבצת השלישית.

int algorithm\_two(char mat[3][3], char me, char he,int i,int j, int column, int row, int main\_diagonal, int secondary\_diagonal,

int columnx, int rowx, int main\_diagonalx, int secondary\_diagonalx) {

if (row == 2 && rowx == 0) { if (mat[i][0] == 0) { mat[i][0] = me; } if (mat[i][1] == 0) { mat[i][1] = me; } if (mat[i][2] == 0) { mat[i][2] = me; } return 1; }

else if (column == 2 && columnx == 0) {if (mat[0][i] == 0) { mat[0][i] = me; } if (mat[1][i] == 0) { mat[1][i] = me; } if (mat[2][i] == 0) { mat[2][i] = me; } return 1; }

else if (main\_diagonal == 2 && main\_diagonalx == 0) { if (mat[0][0] == 0) { mat[0][0] = me; } if (mat[1][1] == 0) { mat[1][1] = me; } if (mat[2][2] == 0) { mat[2][2] = me; } return 1; }

else if (secondary\_diagonal == 2 && secondary\_diagonalx == 0) { if (mat[0][2] == 0) { mat[0][2] = me; } if (mat[1][1] == 0) { mat[1][1] = me; } if (mat[2][0] == 0) { mat[2][0] = me; } return 1; }

return 0;

}

//חלק 3

//אם בשורה או באלכסון מסומנות שתי משבצות של היריב – לסמן את המשבצת השלישית.

int algorithm\_three(char mat[3][3], char me, char he, int i, int j, int column, int row, int main\_diagonal, int secondary\_diagonal,

int columnx, int rowx, int main\_diagonalx, int secondary\_diagonalx) {

if (rowx == 2 && row == 0) { if (mat[i][0] == 0) { mat[i][0] = me; } if (mat[i][1] == 0) { mat[i][1] = me; } if (mat[i][2] == 0) { mat[i][2] = me; } return 1; }

else if (columnx == 2 && column == 0) { if (mat[0][i] == 0) { mat[0][i] = me; } if (mat[1][i] == 0) { mat[1][i] = me; } if (mat[2][i] == 0) { mat[2][i] = me; } return 1; }

else if (main\_diagonalx == 2 && main\_diagonal == 0) { if (mat[0][0] == 0) { mat[0][0] = me; } if (mat[1][1] == 0) { mat[1][1] = me; } if (mat[2][2] == 0) { mat[2][2] = me; } return 1; }

else if (secondary\_diagonalx == 2 && secondary\_diagonal == 0) { if (mat[0][2] == 0) { mat[0][2] = me; } if (mat[1][1] == 0) { mat[1][1] = me; } if (mat[2][0] == 0) { mat[2][0] = me; } return 1; }

return 0;

}

//חלק 4

//אם יש משבצת הנמצאת על טור/שורה/אלכסון של שתי משבצות אחרות שלי, ואותו הטור/שורה/אלכסונים לא כוללים משבצת יריב - לסמנה.

int algorithm\_four(char mat[3][3], char me, char he ) {//מונה שלו

int i1,i,j;

int columna = 0, rowa = 0, main\_diagonala = 0, secondary\_diagonala = 0;

int counter = 0;

for ( i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

if (mat[i][j] == 0) {

for (i1 = 0; i1 < 3; i1++){

if (mat[i][i1] == me) //שורות

rowa++;

else if (mat[i][i1] == he)

rowa--;

if (rowa >= 2)

rowa = 1;

if (rowa < 0)

rowa = 0;

//עמודות

if (mat[i1][j] == me)

columna++;

else if (mat[i1][j] == he)

columna--;

if (columna >= 2)

columna = 1;

if (columna < 0)

columna = 0;

if (i - j == 0) {//אלכסון ראשי

if (mat[i1][i1] == me)

main\_diagonala++;

else if(mat[i1][i1] == he)

main\_diagonala--;

if(main\_diagonala >= 2)

main\_diagonala = 1;

if (main\_diagonala < 0)

main\_diagonala = 0;

}

if (i + j == 2) {//אלכסון משני

if (mat[i1][2 - i1] == me)

secondary\_diagonala++;

else if (mat[i1][2 - i1] == he)

secondary\_diagonala--;

if (secondary\_diagonala >= 2)

secondary\_diagonala = 1;

if (secondary\_diagonala < 0)

secondary\_diagonala = 0;

}

}

if ((columna + rowa + main\_diagonala + secondary\_diagonala) >= 2) { mat[i][j] = me; return 1; }

columna = 0; rowa = 0; main\_diagonala = 0; secondary\_diagonala = 0;

}

}

}

return 0;

}

//חלק 5

//אם שלוש משבצות מסומנות, המשבצת המרכזית שלי ושתי משבצות מנוגדות מסומנות– לסמן משבצת קצה.

int algorithm\_five(char mat[3][3], char me, char he, int counter) {

int tow = 0;

if (mat[1][1] == me && counter == 3) {

if (mat[0][0] == he) { tow++; }if (mat[2][2] == he) { tow++; }if (mat[0][2] == he) { tow++; }if (mat[2][0] == he) { tow++; }

if (tow == 2) {

if (mat[0][0] == 0) { mat[0][0] = me; }

else if (mat[2][2] == 0) { mat[2][2] = me; }

else if (mat[0][2] == 0) { mat[0][2] = me; }

else if (mat[2][0] == 0) { mat[2][0] = me; }return 1;

}

}

}

//חלק 6

//אם שלוש משבצות מסומנות והמשבצת המרכזית שלי - לסמן פינה הנמצאת באותו טור/שורה עם שתי המשבצות של היריב ביחד.

int algorithm\_six(char mat[3][3], char me, char he,int counter) {//מונה שלו

int i1, i, j;

int columna = 0, rowa = 0;

if (counter == 3 && mat[1][1] == me) {

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

if (mat[i][j] == 0) {

for (i1 = 0; i1 < 3; i1++) {

if (mat[i][i1] == he) //שורות

rowa++;

//עמודות

if (mat[i1][j] == he)

columna++;

}

}

if ((columna + rowa) >= 2) { mat[i][j] = me; return 1; }

columna = 0; rowa = 0;

}

}

}

return 0;

}

//חלק 7

//אם יש פינה פנויה ומשני צדדיה שתי שורות ריקות – לסמן את הפינה.

int algorithm\_seven(char mat[3][3], char me, char he) {

int i, j, row1=0, row2=0, column1=0, column2=0;

if (mat[0][0] == 0 && mat[0][1] == 0 && mat[0][2] == 0) { row1++; }

if (mat[2][0] == 0 && mat[2][1] == 0 && mat[2][2] == 0) { row2++; }

if (mat[0][0] == 0 && mat[1][0] == 0 && mat[2][0] == 0) { column1++; }

if (mat[0][2] == 0 && mat[1][2] == 0 && mat[2][2] == 0) { column2++; }

if (row1 + column1 == 2) { mat[0][0] = me; return 1; } else if (row1 + column2 == 2) { mat[0][2] = me; return 1; }

else if (row2 + column2 == 2){ mat[2][2] = me; return 1; } else if (row2 + column1 == 2) { mat[2][0] = me; return 1; }

return 0;

}

//חלק 8

int algorithm\_eight(char mat[3][3], char me, char he) {

if (mat[1][1] == 0) { mat[1][1] = me; return 1; }

return 0;

}

//חלק 9

//אם יש שלוש משבצות פנויות בשורה או באלכסון – לסמן את הפינה.

int algorithm\_nine(char mat[3][3], char me, char he,int i,int j,

int column,int row,int main\_diagonal ,int secondary\_diagonal ,

int columnx ,int rowx ,int main\_diagonalx ,int secondary\_diagonalx ){

if (column + columnx == 0) { mat[0][i] = me; return 1; }

else if (row + rowx == 0) { mat[i][0] = me; return 1; }

else if (main\_diagonal + main\_diagonalx == 0) { mat[2][2] = me; return 1; }

else if (secondary\_diagonal + secondary\_diagonalx == 0) { mat[0][2] = me; return 1; }

return 0;

}

//חלק 10

// אם יש שתי משבצות מנוגדות והמשבצת האמצעית שלי – לסמן אחת משתיים

int algorithm\_ten(char mat[3][3], char me, char he,int i,

int column, int row, int main\_diagonal, int secondary\_diagonal,

int columnx ,int rowx ,int main\_diagonalx,int secondary\_diagonalx ) {

if (mat[1][1] == me) {

if (mat[0][1] == 0 && mat[2][1] == 0) { mat[2][1] = me; return 1; }

if (mat[1][0] == 0 && mat[1][2] == 0) { mat[1][0] = me; return 1; }

if (main\_diagonal == 1 && main\_diagonalx==0) { mat[2][2] = me; return 1; }

if (secondary\_diagonal == 1 && secondary\_diagonalx==0) { mat[2][2] = me; return 1; }

}

return 0;

}

//חלק 11

// אם יש משבצת פינתית פנויה - לסמן אותה.

intalgorithm\_eleven(char mat[3][3], char me, char he) {

if (mat[0][0] == 0) { mat[0][0] = me; return 1; }if (mat[0][2] == 0) { mat[0][2] = me; return 1; }

if (mat[2][0] == 0) { mat[2][0] = me; return 1; }if (mat[2][2] == 0) { mat[2][2] = me; return 1; }

return 0;

}

//חלק 12

// אם יש משבצת פנויה - לסמן אותה.

int intalgorithm\_twelve(char mat[3][3], char me, char he) {

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

if (mat[i][j]==0){

mat[i][j] = me;

return 1;

}

}

}

return 0;

}

//אלגוריתם

int algorithm\_x\_o(char mat[3][3], char me, char he) {

int column = 0, row = 0, main\_diagonal = 0, secondary\_diagonal = 0;

int columnx = 0, rowx = 0, main\_diagonalx = 0, secondary\_diagonalx = 0;

int counter = 0;

int i, j, x;

for (x = 0; x < 12; x++){

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

if (mat[i][j] == 'x' || mat[i][j] == 'o')//כללי

counter++;

if (mat[i][j] == 'o')//שורות

row++;

else if (mat[i][j] == 'x')

rowx++;

if (mat[j][i] == 'o')//עמודות

column++;

else if (mat[j][i] == 'x')

columnx++;

if (mat[j][j] == 'o')//אלכסון ראשי

main\_diagonal++;

else if (mat[j][j] == 'x')

main\_diagonalx++;

if (mat[j][2 - j] == 'o')//אלכסון משני

secondary\_diagonal++;

else if (mat[j][2 - j] == 'x')

secondary\_diagonalx++;

if (x==0&&algorithm\_one(mat, me, he, counter) == 1) {//חלק 1

return 1;

}

}

if (x==1&&algorithm\_two(mat, me, he, i, j, column, row, main\_diagonal, secondary\_diagonal,//חלק 2

columnx, rowx, main\_diagonalx, secondary\_diagonalx) == 1) {

return 1;

}

if (x==2&&algorithm\_three(mat, me, he, i, j, column, row, main\_diagonal, secondary\_diagonal,//חלק 3

columnx, rowx, main\_diagonalx, secondary\_diagonalx) == 1) {

return 1;

}

if (x==3&&algorithm\_four(mat, me, he) == 1) { return 1; }//חלק 4

if (x==4&&algorithm\_five(mat, me, he, counter) == 1) { return 1; }//חלק 5

if (x==5&&algorithm\_six(mat, me, he, counter) == 1) { return 1; }//חלק 6

if (x==6&&algorithm\_seven(mat, me, he) == 1) { return 1; }//חלק 7

if (x==7&&algorithm\_eight(mat, me, he) == 1) { return 1; }//חלק 8

if (x==8 &&algorithm\_nine(mat, me, he, i, j,

column, row, main\_diagonal, secondary\_diagonal,//חלק 9

columnx, rowx, main\_diagonalx, secondary\_diagonalx) == 1) {

return 1;

}

if (x==9 &&algorithm\_ten(mat, me, he, i,

column, row, main\_diagonal, secondary\_diagonal,//חלק 10

columnx, rowx, main\_diagonalx, secondary\_diagonalx) == 1) {

return 1;

}

if (x==10&&intalgorithm\_eleven(mat, me, he) == 1) { return 1; }//חלק 11

if (x==11&&intalgorithm\_twelve(mat, me, he) == 1) { return 1; }

column = 0; row = 0; main\_diagonal = 0; secondary\_diagonal = 0;

columnx = 0; rowx = 0; main\_diagonalx = 0; secondary\_diagonalx = 0;

}

}

return 0;

}

void the\_game() {

char mat[3][3] = { 0 };

char me = 'o', he = 'x';

int turn, flag = 1;

turn = strat\_x\_o();

location\_x\_o();

print(mat);

while (flag)

{

while (turn%2==0)

{

if (winner\_test(mat) == 1) {

printf("the O won ");

flag = 0;

}

else if (winner\_test(mat) == 2) {

printf("the X won ");

flag = 0;

}

else if (winner\_test(mat) == 3) {

printf("the game and in drao");

flag = 0;

}

scan\_x\_o(mat, he);

turn++;

}

while (turn % 2 == 1&& flag==1)

{

algorithm\_x\_o(mat, me, he);

print(mat);

if (winner\_test(mat) == 1) {

printf("\nthe O won \n\n");

flag = 0;

}

if (winner\_test(mat) == 2) {

printf("\n\nthe X won \n\n");

flag = 0;

}

else if (winner\_test(mat) == 3) {

printf("the game and in drao");

flag = 0;

}

turn++;

}

}

}

void main() {

the\_game();

}