/\* A simple Tic Tac Toe game. \*/ #include <stdio.h> #include <stdlib.h>

char matrix[3][3]; /\* the tic tac toe matrix \*/

char check(void); void init\_matrix(void); void get\_player\_move(void); void get\_computer\_move(void); void disp\_matrix(void);

int main(void)

{ char done;

printf("This is the game of Tic Tac Toe.\n"); printf(''You will be playing against the computer.\n");

done = ' '; init\_matrix();

do { disp\_matrix(); get\_player\_move(); done = check(); /\* see if winner \*/ if(done!= ' ') break; /\* winner!\*/ get\_computer\_move (); done = check(); /\* see if winner \*/ } while(done== ' ');

if(done=='X') printf("You won!\n"); else printf("I won!!!!\n"); disp\_matrix(); /\* show final positions \*/

return 0; }

/\* Initialize the matrix. \*/ void init\_matrix(void) { int i, j;

for(i=0; i<3; i++) for(j=0; j<3; j++) matrix[i][j] = ' '; }

/\* Get a player's move. \*/ void get\_player\_move (void) { int x, y;

printf("Enter X,Y coordinates for your move: ");

scanf("%d%\*c%d", &x, &y);

x--; y--;

if(matrix[x][y]!= ' '){ printf(''Invalid move, try again.\n"); get\_player\_move(); } else matrix[x][y] = 'X'; }

/\* Get a move from the computer. \*/ void get\_computer\_move(void) { int i, j; for(i=0; i<3; i++){ for(j=0; j<3; j++) if(matrix[i][j]==' ') break; if(matrix[i][j]==' ') break; }

if(i\*j==9) { printf("draw\n"); exit(0); } else matrix[i][j] = 'O'; }

/\* Display the matrix on the screen. \*/ void disp\_matrix(void) { int t;

for(t=0; t<3; t++) { printf(" %c | %c | %c ",matrix[t][0], matrix[t][1], matrix [t][2]); if(t!=2) printf("\n---|---|---\n"); } printf ( "\n"); }

/\* See if there is a winner. \*/ char check(void)

{ int i;

for(i=0; i<3; i++) /\* check rows \*/ if(matrix[i][0]==matrix[i][1] && matrix[i][0]==matrix[i][2]) return matrix[i][0];

for(i=0; i<3; i++) /\* check columns \*/ if(matrix[0][i]==matrix[1][i] && matrix[0][i]==matrix[2][i]) return matrix[0] [i];

/\* test diagonals \*/ if(matrix[0] [0]==matrix[1] [1] && matrix[1][1]==matrix[2][2]) return matrix[0][0];

if(matrix[0] [2]==matrix[1] [1] && matrix[1] [1]==matrix[2][0]) return matrix[0][2];

return ' ';

}