

مینا نسیم ادوار نسیم یوسف

سیکشن 7

ثالثہ حاسبات ونظم تحكم

```
#include <stdio.h>

#define EMPTY ' '

#define PLAYER_X 'X'

#define PLAYER_O 'O'

char grid[3][3] = {
    {EMPTY, EMPTY, EMPTY},
    {EMPTY, EMPTY, EMPTY},
    {EMPTY, EMPTY, EMPTY},
};

int isGameOver(char grid[][3]) {
    // Check rows
    for (int i = 0; i < 3; i++) {
        if (grid[i][0] == grid[i][1] && grid[i][1] == grid[i][2] && grid[i][0] != EMPTY) {
            return 1; // Winner found
        }
    }

    // Check columns
    for (int j = 0; j < 3; j++) {
        if (grid[0][j] == grid[1][j] && grid[1][j] == grid[2][j] && grid[0][j] != EMPTY) {
            return 1; // Winner found
        }
    }

    // Check diagonals
    if (grid[0][0] == grid[1][1] && grid[1][1] == grid[2][2] && grid[0][0] != EMPTY) {
        return 1; // Winner found
    }
}
```

```

}

if (grid[0][2] == grid[1][1] && grid[1][1] == grid[2][0] && grid[0][2] != EMPTY) {
    return 1; // Winner found
}

// Check for draw (all cells filled)
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        if (grid[i][j] == EMPTY) {
            return 0; // Game not over, empty cells present
        }
    }
}

return 2; // Draw
}

```

```

void printGrid(char grid[][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%c ", grid[i][j]);
        }
        printf("\n");
    }
}

```

```

int main() {
    char currentPlayer = PLAYER_X;
    int gameOver = 0;
    while (!gameOver) {
        printGrid(grid);
        printf("Turn: %c\n", currentPlayer);
    }
}

```

```
int index;

// Get valid input from the user
do {
printf("Enter an index (1-9): ");
scanf("%d", &index);
} while (index < 1 || index > 9 || grid[(index - 1) / 3][(index - 1) % 3] != EMPTY);

// Decrement index to match array indexing (0-based)
index--;

// Update grid
grid[index / 3][index % 3] = currentPlayer;

gameOver = isGameOver(grid);

if (gameOver == 1) {
printGrid(grid);
printf("Player %c wins!\n", currentPlayer);
} else if (gameOver == 2) {
printf("It's a draw!\n");
printf("Starting new game...\n");

// Reset grid
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
grid[i][j] = EMPTY;
}
}
}

// Switch player for next turn
currentPlayer = (currentPlayer == PLAYER_X) ? PLAYER_O : PLAYER_X;
}

return 0;
}
```

```

1  #include <stdio.h>
2  #define EMPTY ' '
3  #define PLAYER_X 'X'
4  #define PLAYER_O 'O'
5  char grid[3][3] = {
6      {EMPTY, EMPTY, EMPTY},
7      {EMPTY, EMPTY, EMPTY},
8      {EMPTY, EMPTY, EMPTY},
9  };
10 int isGameOver(char grid[][3]) {
11     // Check rows
12     for (int i = 0; i < 3; i++) {
13         if (grid[i][0] == grid[i][1] && grid[i][1] == grid[i][2] && grid[i][0] != EMPTY) {
14             return 1; // Winner found
15         }
16     }
17     // Check columns
18     for (int j = 0; j < 3; j++) {
19         if (grid[0][j] == grid[1][j] && grid[1][j] == grid[2][j] && grid[0][j] != EMPTY) {
20             return 1; // Winner found
21         }
22     }
23     // Check diagonals
24     if (grid[0][0] == grid[1][1] && grid[1][1] == grid[2][2] && grid[0][0] != EMPTY) {
25         return 1; // Winner found
26     }
27     if (grid[0][2] == grid[1][1] && grid[1][1] == grid[2][0] && grid[0][2] != EMPTY) {
28         return 1; // Winner found
29     }
30     // Check for draw (all cells filled)
31     for (int i = 0; i < 3; i++) {
32         for (int j = 0; j < 3; j++) {
33             if (grid[i][j] == EMPTY) {
34                 return 0; // Game not over, empty cells present
35             }
36         }
37     }
38     return 2; // Draw
39 }
40
41 void printGrid(char grid[][3]) {
42     for (int i = 0; i < 3; i++) {
43         for (int j = 0; j < 3; j++) {
44             printf("%c ", grid[i][j]);
45         }
46         printf("\n");
47     }
48 }
49
50 int main() {
51     char currentPlayer = PLAYER_X;
52     int gameOver = 0;
53     while (!gameOver) {
54         printGrid(grid);
55         printf("Turn: %c\n", currentPlayer);
56         int index;
57         // Get valid input from the user
58         do {
59             printf("Enter an index (1-9): ");
60             scanf("%d", &index);
61         } while (index < 1 || index > 9 || grid[(index - 1) / 3][(index - 1) % 3] != EMPTY);
62         // Decrement index to match array indexing (0-based)
63         index--;
64         // Update grid
65         grid[index / 3][index % 3] = currentPlayer;
66         gameOver = isGameOver(grid);
67         if (gameOver == 1) {
68             printGrid(grid);
69             printf("Player %c wins!\n", currentPlayer);
70         } else if (gameOver == 2) {
71             printf("It's a draw!\n");
72             printf("Starting new game...\n");
73             // Reset grid
74             for (int i = 0; i < 3; i++) {
75                 for (int j = 0; j < 3; j++) {
76                     grid[i][j] = EMPTY;
77                 }
78             }
79             // Switch player for next turn
80             currentPlayer = (currentPlayer == PLAYER_X) ? PLAYER_O : PLAYER_X;
81         }
82     }
83     return 0;
84 }

```

Text logs from terminal when the program is working in the following condition

■ (X) win the game

Turn: X

Enter an index (1-9): 1

X

Turn: O

Enter an index (1-9): 2

X O

Turn: X

Enter an index (1-9): 4

X O

X

Turn: O

Enter an index (1-9): 3

X O O

X

Turn: X

Enter an index (1-9): 7

X O O

X

X

Player X wins!

■ (O) win the game

Turn: X

Enter an index (1-9): 2

X

Turn: O

Enter an index (1-9): 3

X O

Turn: X

Enter an index (1-9): 1

X X O

Turn: O

Enter an index (1-9): 5

X X O

O

Turn: X

Enter an index (1-9): 6

X X O

O X

Turn: O

Enter an index (1-9): 7

X X O

O X

O

Player O wins!

■ (X) win the game after 2 draw condition.

Turn: X

Enter an index (1-9): 1

X

Turn: O

Enter an index (1-9): 4

X

O

Turn: X

Enter an index (1-9): 3

X X

O

Turn: O

Enter an index (1-9): 7

X X

O

O

Turn: X

Enter an index (1-9): 5

X X

O X

O

Turn: O

Enter an index (1-9): 6

X X

O X O

O

Turn: X

Enter an index (1-9): 8

X X

O X O

O X

Turn: O

Enter an index (1-9): 9

X X

O X O

O X O

Turn: X

Enter an index (1-9): 2

X X X

O X O

O X O

Player X wins!

■ (O) wins the game after 1 draw condition.

Turn: X

Enter an index (1-9): 1

X

Turn: O

Enter an index (1-9): 2

X O

Turn: X

Enter an index (1-9): 3

X O X

Turn: O

Enter an index (1-9): 5

X O X

O

Turn: X

Enter an index (1-9): 6

X O X

O X

Turn: O

Enter an index (1-9): 8

X O X

O X

O

Player O wins!