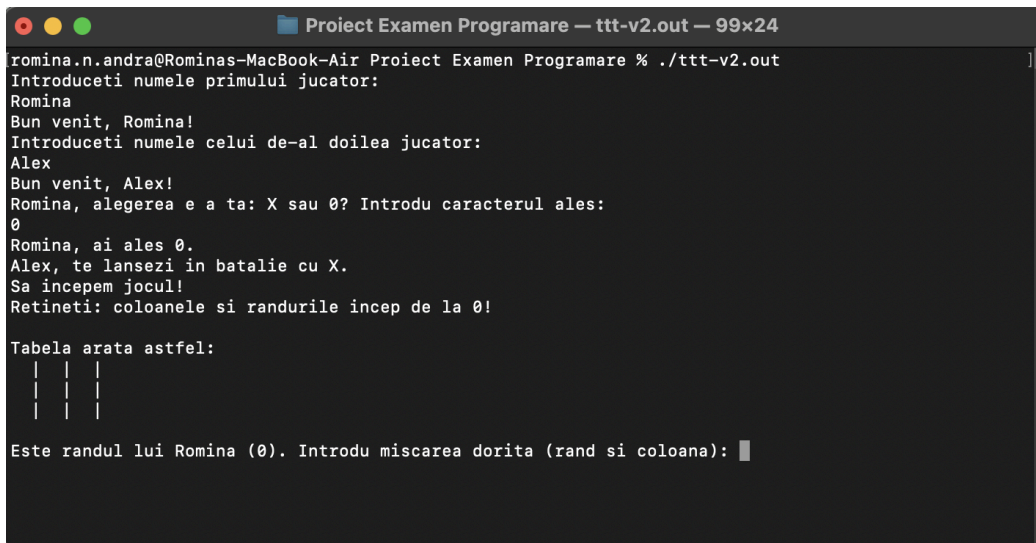


Documentație Program X și O

Program în Limbaj C

Descriere Program

Acest program transpune jocul clasic de X și O (Tic-Tac-Toe) în limbajul C. Pe rând, jucătorii își introduc numele și aleg fiecare între simbolurile X și O. Jocul se desfășoară pe o tablă de 3x3 și continuă până când un jucător câștigă sau tabla este plină, fără un câștigător.



```
romina.n.andra@Rominas-MacBook-Air Proiect Examen Programare % ./ttt-v2.out
Introduceti numele primului jucator:
Romina
Bun venit, Romina!
Introduceti numele celui de-al doilea jucator:
Alex
Bun venit, Alex!
Romina, alegerea e a ta: X sau 0? Introdu caracterul ales:
0
Romina, ai ales 0.
Alex, te lansezi in batalie cu X.
Sa incepem jocul!
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
| | |
| | |
| | |

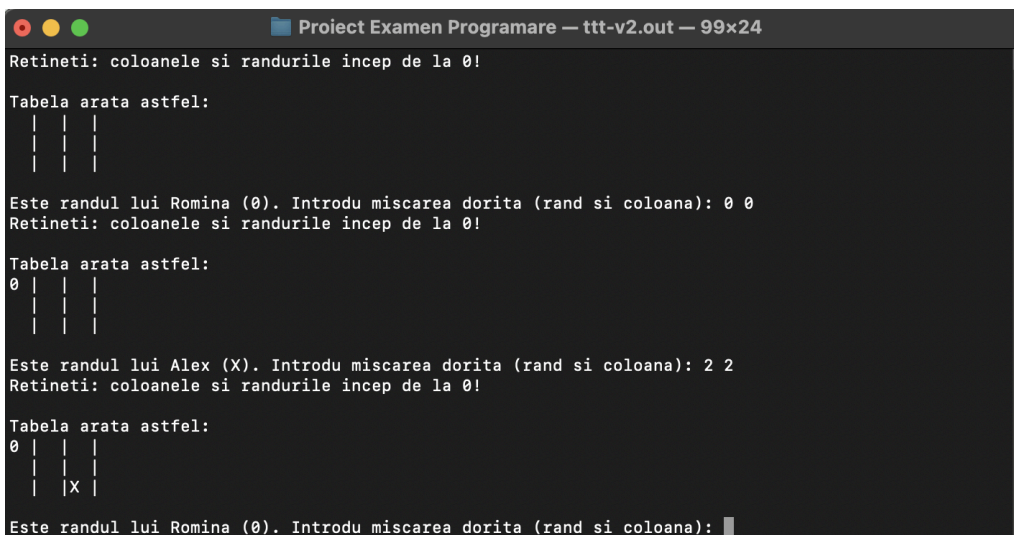
Este randul lui Romina (0). Introdu miscarea dorita (rand si coloana):
```

Utilizare

Programul se compilează și execută în terminal.

Introduceți numele jucătorilor și alegeți fiecare simbolul (X sau O) prin tasta Enter.

Mai apoi, introduceți coordonatele (rândul și coloana) pentru a face mișcările pe tabla de joc.



```
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
| | |
| | |
| | |

Este randul lui Romina (0). Introdu miscarea dorita (rand si coloana): 0 0
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
0 | | |
| | |
| | |

Este randul lui Alex (X). Introdu miscarea dorita (rand si coloana): 2 2
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
0 | | |
| | |
| |X|

Este randul lui Romina (0). Introdu miscarea dorita (rand si coloana):
```

Numărătoarea coordonatelor (rândul și coloana) începe de la zero (0) în loc de 1. Acestea pot fi introduse cu spațiu între ele urmate de Enter, ori prin simpla apăsare a tastei Enter după fiecare coordonată.

```
Proiect Examen Programare — ttt-v2.out — 99x24

Tabela arata astfel:
0 |X| |
  |0| |
  | | |

Este randul lui Alex (X). Introdu miscarea dorita (rand si coloana): 2 2
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
0 |X| |
  |0| |
  |X| |

Este randul lui Romina (0). Introdu miscarea dorita (rand si coloana): 1
0
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
0 |X| |
0 |0| |
  |X| |

Este randul lui Alex (X). Introdu miscarea dorita (rand si coloana):
```

Jocul se termină când un jucător câștigă sau tabla este completă, ducând la egalitate.

```
Proiect Examen Programare — -zsh — 99x24

Tabela arata astfel:
|0|0|
|X|0|
|X|X|

Este randul lui Romina (0). Introdu miscarea dorita (rand si coloana): 1 0
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
|0|0|
0|X|0|
|X|X|

Este randul lui Alex (X). Introdu miscarea dorita (rand si coloana): 0 0
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
X|0|0|
0|X|0|
|X|X|

X a castigat! Felicitari!
romina.n.andra@Rominas-MacBook-Air Proiect Examen Programare %
```

```
Proiect Examen Programare — -zsh — 99x24

Tabela arata astfel:
0|X|0|
0|0|X|
| |X|

Este randul lui Alex (X). Introdu miscarea dorita (rand si coloana): 2 0
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
0|X|0|
0|0|X|
X| |X|

Este randul lui Romina (0). Introdu miscarea dorita (rand si coloana): 2 1
Retineti: coloanele si randurile incep de la 0!

Tabela arata astfel:
0|X|0|
0|0|X|
X|0|X|

Egalitate! Incercam din nou?
romina.n.andra@Rominas-MacBook-Air Proiect Examen Programare %
```

Codul Programului

Structura *Player*

Se definește structura care va conține numele jucătorului și simbolul acestuia (X sau O).

Sunt create trei variabile de tipul acestei structuri: Player1, Player2, PlayerCurrent.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  struct Player {
5      char Nume[50];
6      char XorO;
7  } Player1, Player2, PlayerCurrent;
8
```

Funcții

NumeJucatori()

Afișează mesajele de început și cere utilizatorului să introducă datele pentru a începe jocul.

Mai apoi, stochează numele jucătorilor în variabilele Player1.Nume și Player2.Nume.

Xsau0Jucatori()

Afișează mesaje pentru alegerea simbolului (X sau O) pentru primul jucător.

Ulterior, setează simbolul jucătorului 2 în funcție de alegerea jucătorului 1.

```
8
9  void NumeJucatori() {
10     printf("Introduceti numele primului jucator:\n");
11     scanf("%s", Player1.Nume);
12     printf("Bun venit, %s!\n", Player1.Nume);
13     printf("Introduceti numele celui de-al doilea jucator:\n");
14     scanf("%s", Player2.Nume);
15     printf("Bun venit, %s!\n", Player2.Nume);
16 }
17
18 void Xsau0Jucatori() {
19     printf("%s, alegerea e a ta: X sau O? Introdu caracterul ales:\n", Player1.Nume);
20     scanf(" %c", &Player1.XorO);
21     printf("%s, ai ales %c.\n", Player1.Nume, Player1.XorO);
22     if (Player1.XorO == 'X') {
23         Player2.XorO = 'O';
24     } else {
25         Player2.XorO = 'X';
26     }
27     printf("%s, te lansezi in batalie cu %c.\n", Player2.Nume, Player2.XorO);
28     printf("Sa incepem jocul!\n");
29 }
30
```

AfisareBoard(char board[3][3])

Creează și afișează tabla de joc cu elementele curente, marcate prin spații și linii.

ConditiiCastig(char board[3][3], char Xor0)

Stabilește și verifică dacă există condiții de câștig pe orizontală, verticală sau diagonală pentru simbolul dat (X sau 0).

```
30
31 void AfisareBoard(char board[3][3]) {
32     printf("Retineti: coloanele si randurile incep de la 0!\n");
33     printf("\n");
34     printf("Tabela arata astfel: \n");
35     for (int i = 0; i < 3; ++i) {
36         for (int j = 0; j < 3; ++j) {
37             printf("%c |", board[i][j]);
38         }
39         printf("\n");
40     }
41     printf("\n");
42 }
43
44 int ConditiiCastig(char board[3][3], char Xor0) {
45     for (int i = 0; i < 3; ++i) {
46         if ((board[i][0] == Xor0 && board[i][1] == Xor0 && board[i][2] == Xor0) ||
47             (board[0][i] == Xor0 && board[1][i] == Xor0 && board[2][i] == Xor0)) {
48             return 1;
49         }
50     }
51
52     if ((board[0][0] == Xor0 && board[1][1] == Xor0 && board[2][2] == Xor0) ||
53         (board[0][2] == Xor0 && board[1][1] == Xor0 && board[2][0] == Xor0)) {
54         return 1;
55     }
56
57     return 0;
58 }
59
```

BoardFull(char board[3][3])

Stabilește și verifică dacă tabla de joc este completă (fără spații goale).

copiazaPlayerCurent(struct Player player)

Copiază numele și simbolul jucătorului dat în variabila PlayerCurrent, pentru a face schimbul între cei doi jucatori pe rând în timpul jocului.

```
59
60 int BoardFull(char board[3][3]) {
61     for (int i = 0; i < 3; ++i) {
62         for (int j = 0; j < 3; ++j) {
63             if (board[i][j] == ' ') {
64                 return 0;
65             }
66         }
67     }
68     return 1;
69 }
70
71 void copiazaPlayerCurent(struct Player player) {
72     for (int i = 0; i < 50; i++) {
73         PlayerCurrent.Nume[i] = player.Nume[i];
74     }
75     PlayerCurrent.Xor0 = player.Xor0;
76 }
77
```

Funcția Principală (main)

Folosindu-se de funcțiile anterioare, afișează mesajele, tabla de joc și variabilele de jucători.

Odată ce informațiile despre jucători au fost introduse, atâta timp cât condițiile de sfârșit de joc nu sunt îndeplinite sau tabla este plină, programul continuă să ruleze.

La fiecare repetare, programul afișează tabla de joc, solicită mișcarea jucătorului curent prin coordonate și verifică condițiile mișcării.

După încheierea jocului, afișează rezultatul (câștigător sau egalitate).

```
78 int main() {
79     NumeJucatori();
80     XsauOJucatori();
81
82     char board[3][3] = {
83         {' ', ' ', ' '},
84         {' ', ' ', ' '},
85         {' ', ' ', ' '},
86     };
87
88     int row, col;
89     copiazaPlayerCurrent(Player1);
90
91     while (!ConditiiCastig(board, 'X') && !ConditiiCastig(board, 'O') && !BoardFull(board)) {
92         AfișareBoard(board);
93         printf("Este randul lui %s (%c). Introdu mișcarea dorită (rand și coloana): ", PlayerCurrent.Nume, PlayerCurrent.XorO);
94         scanf("%d %d", &row, &col);
95
96         if (row < 0 || row >= 3 || col < 0 || col >= 3 || board[row][col] != ' ') {
97             printf("Mișcare nereușită. Încearcă din nou.\n");
98         } else {
99             board[row][col] = PlayerCurrent.XorO;
100             if (strcmp(PlayerCurrent.Nume, Player1.Nume) == 0) {
101                 copiazaPlayerCurrent(Player2);
102             }
103             else {
104                 copiazaPlayerCurrent(Player1);
105             }
106         }
107     }
108
109     AfișareBoard(board);
110
111     if (ConditiiCastig(board, 'X')) {
112         printf("X a câștigat! Felicitări!\n");
113     } else if (ConditiiCastig(board, 'O')) {
114         printf("O a câștigat! Felicitări!\n");
115     } else {
116         printf("Egalitate! Încearcă din nou?\n");
117     }
118
119     return 0;
120 }
```

Observații

Funcția *ConditiiCastig* verifică toate direcțiile posibile de victorie (orizontală și diagonală).

După fiecare mișcare, jucătorul curent este actualizat pentru a permite alternarea între jucători prin variabila *PlayerCurrent*.

----- Final -----