OCTOBER 2021 | COMPUTER SCIENCE

COMPUTER ARCHITECTURE
ASSIGNMENT REPORT

TIC-TAC-TOE GAME

TRẦN QUỐC BẢO
2052038

CONTENT

INTRODUCTION

- Tic-Tac-Toe is a paper-and-pencil game for two players who takes turns marking the space in a three-by-three grid with X or O. The player who succeeds in placing three of their mark in a horizontal, vertical, or diagonal row is the winner.

- The simplicity of Tic-Tac-Toe has made it a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the searching of game trees.

- In this assignment, I will design and write MIPS assembly language for implementing a text-based Tic-Tac-Toe game for two players.

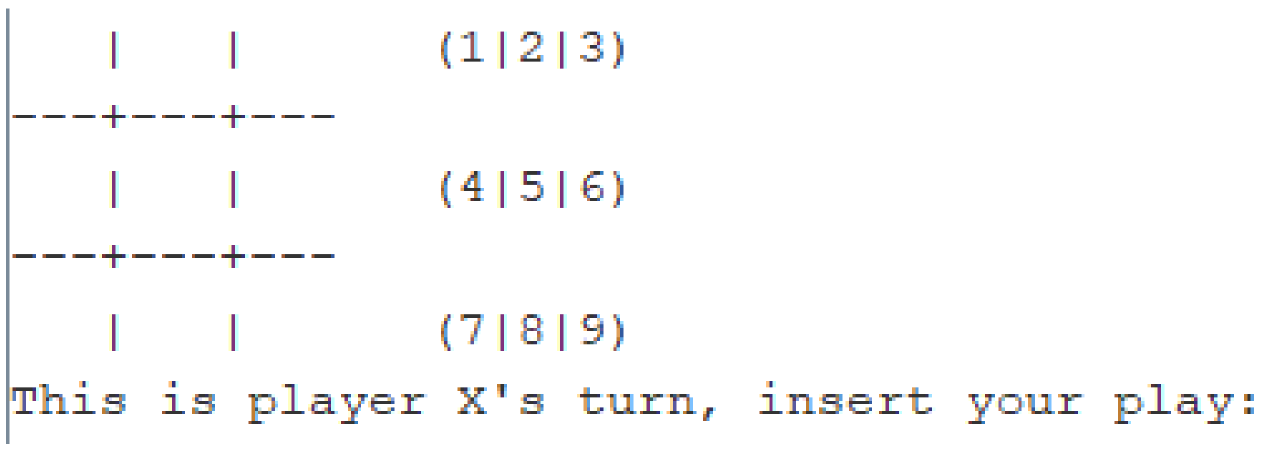
ALGORITHMS AND CODE

Idea

- The main idea was to print out a board initially and each time a move is done with clear visual as user interface.
- Every time the user select a move, the symbol is inserted into the board based on the selected move and which player's turn.
- After each insertion, the program will check if either player has won the game.
- If the whole board has been filled with symbol without any winner yet, the game is tie.
- A menu is printed out requesting user choice (either exist or continue a new game) after each match over.

Game started

- The main function will store the play board in **$s1**, set all the register used in the program to 0 and reset the play board.
- First, let print out the board



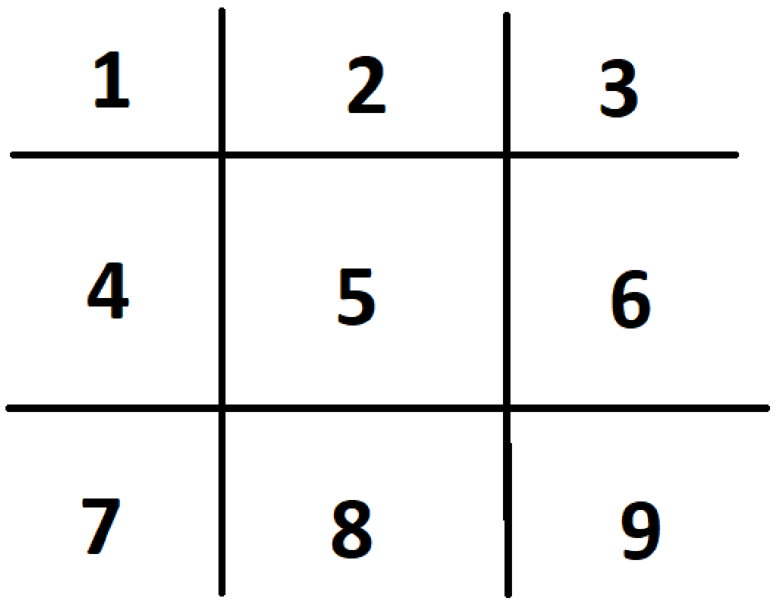
```
   |   |      (1|2|3)
---+---+---
   |   |      (4|5|6)
---+---+---
   |   |      (7|8|9)
This is player X's turn, insert your play:
```

- A value will be used to check for Tie by counting number of move (9 means Tie).

- Otherwise, use it to check which player's turn by checking odd or even number and store the result in **$t0**.

User input

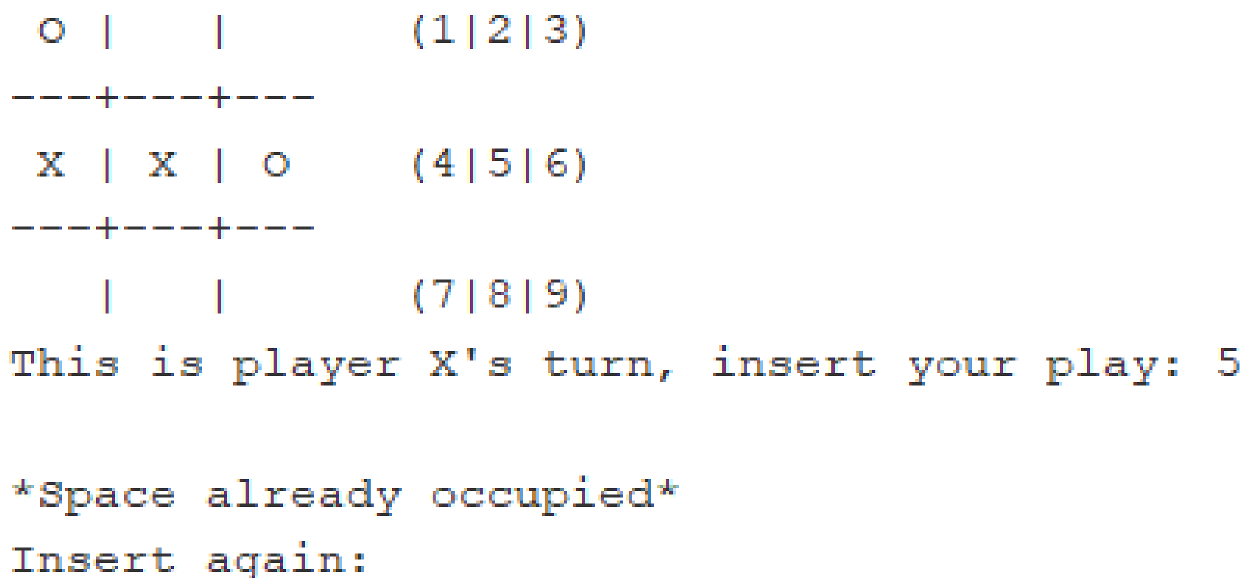
- Player input their move, store in **$s2**.
- The value will be used to determined which move will be inserted and call the respective function.



- Invalid input (out of range 1 to 9) will trigger the invalid function.
- The invalid function print out a text ask for reinput and jump back to the input stage.

Move insertion

- The function would first check if the space is already occupied.



```
 O |   |         (1|2|3)
---+---+---
 X | X | O       (4|5|6)
---+---+---
   |   |         (7|8|9)
This is player X's turn, insert your play: 5

*Space already occupied*
Insert again:
```

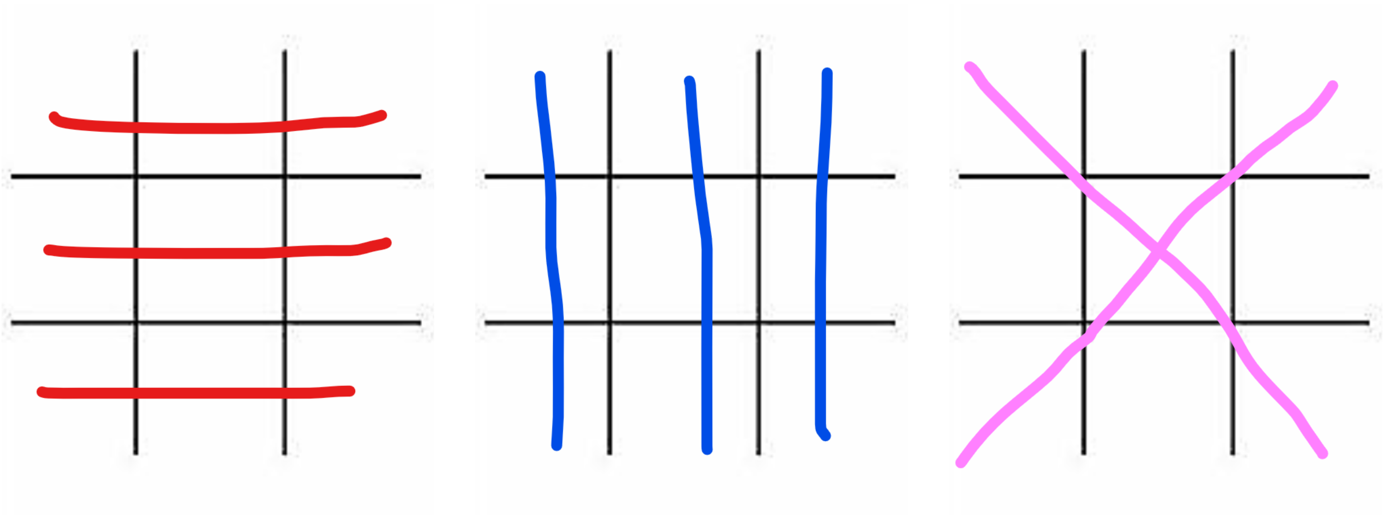
- If not, it will continue to insert symbol to the board. Register **$t1** to **$t9** (depend on move) will store either value 1 or 2 to mark player's turn for checking winner.

- If the space is already occupied, the program will jump back to the input stage.

Victory check

- Which player is the winner will be decided by the value of all the previous register that got assigned to value 1 or 2.

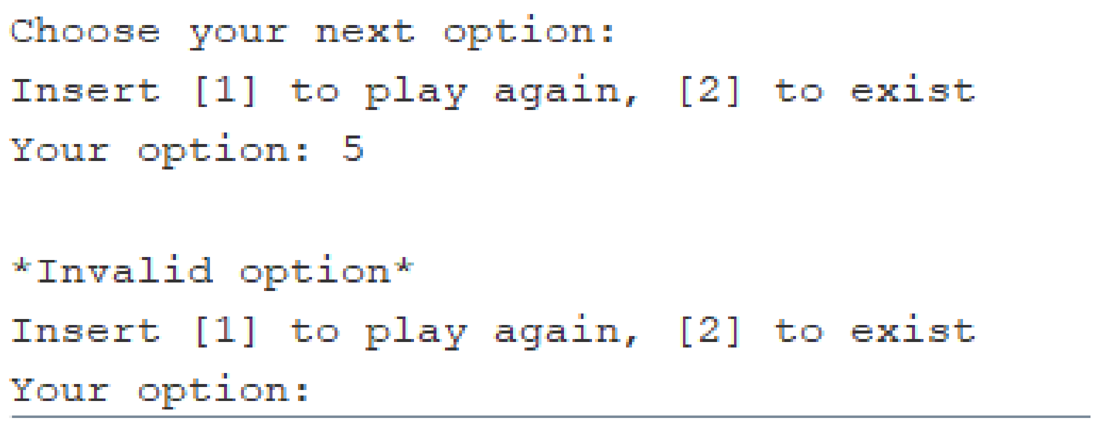
- The function will check both horizontal, vertical and diagonal as the game rule stated using **and** instruction. If one of the three value checked was either all 1 or 2, player X or O respectively will be the winner.



Winner announcement and menu

- The board will be printed out the last time and the player who won the game is announced!

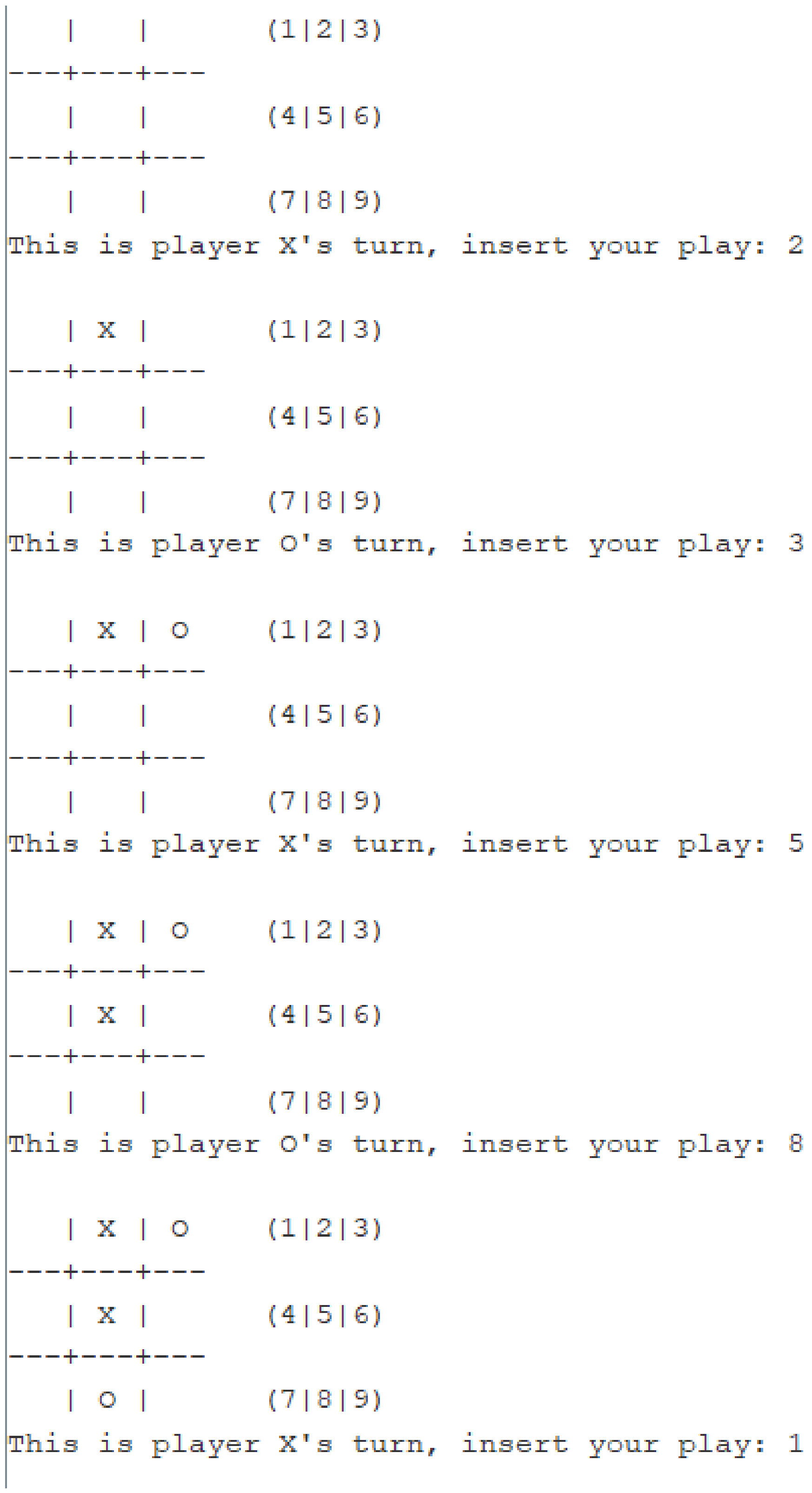
- The program then print out a menu tell the user to choose whether or not they want to start a new game. If a new game is chosen, the program goes back to main, otherwise, the program exist.



```
Choose your next option:
Insert [1] to play again, [2] to exist
Your option: 5

*Invalid option*
Insert [1] to play again, [2] to exist
Your option:
```

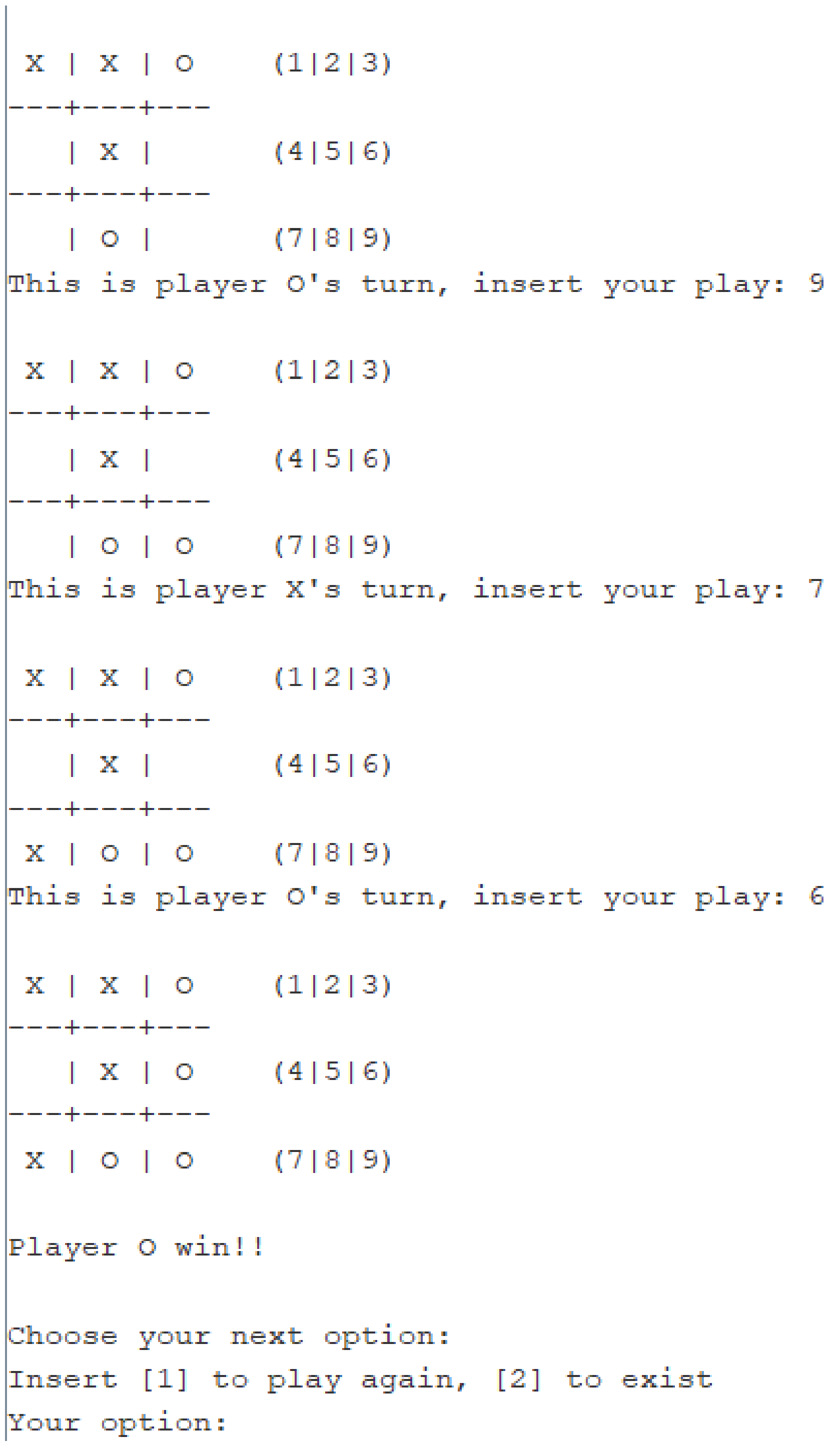
- "Invalid option" text will be printed out and the user need to reinput if the option is invalid.

GAMEPLAY EXAMPLE



```
   |   |        (1|2|3)
---+---+---
   |   |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player X's turn, insert your play: 2

   | X |        (1|2|3)
---+---+---
   |   |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player O's turn, insert your play: 3

   | X | O      (1|2|3)
---+---+---
   |   |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player X's turn, insert your play: 5

   | X | O      (1|2|3)
---+---+---
   | X |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player O's turn, insert your play: 8

   | X | O      (1|2|3)
---+---+---
   | X |        (4|5|6)
---+---+---
   | O |        (7|8|9)
This is player X's turn, insert your play: 1
```

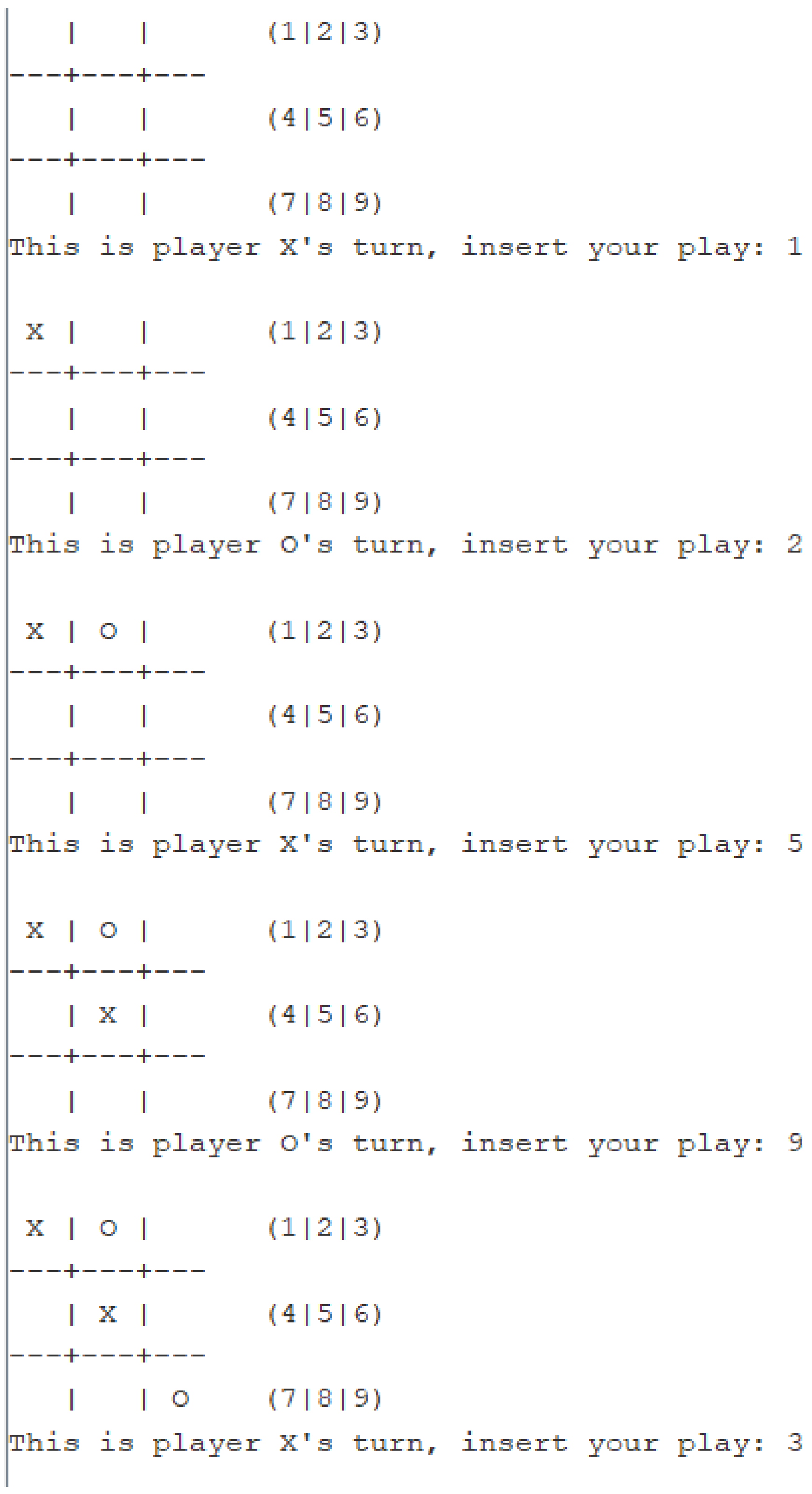
Gameplay Example



```
 X | X | O    (1|2|3)
---+---+---
   | X |      (4|5|6)
---+---+---
   | O |      (7|8|9)
This is player O's turn, insert your play: 9

 X | X | O    (1|2|3)
---+---+---
   | X |      (4|5|6)
---+---+---
   | O | O    (7|8|9)
This is player X's turn, insert your play: 7

 X | X | O    (1|2|3)
---+---+---
   | X |      (4|5|6)
---+---+---
 X | O | O    (7|8|9)
This is player O's turn, insert your play: 6

 X | X | O    (1|2|3)
---+---+---
   | X | O    (4|5|6)
---+---+---
 X | O | O    (7|8|9)

Player O win!!

Choose your next option:
Insert [1] to play again, [2] to exist
Your option:
```

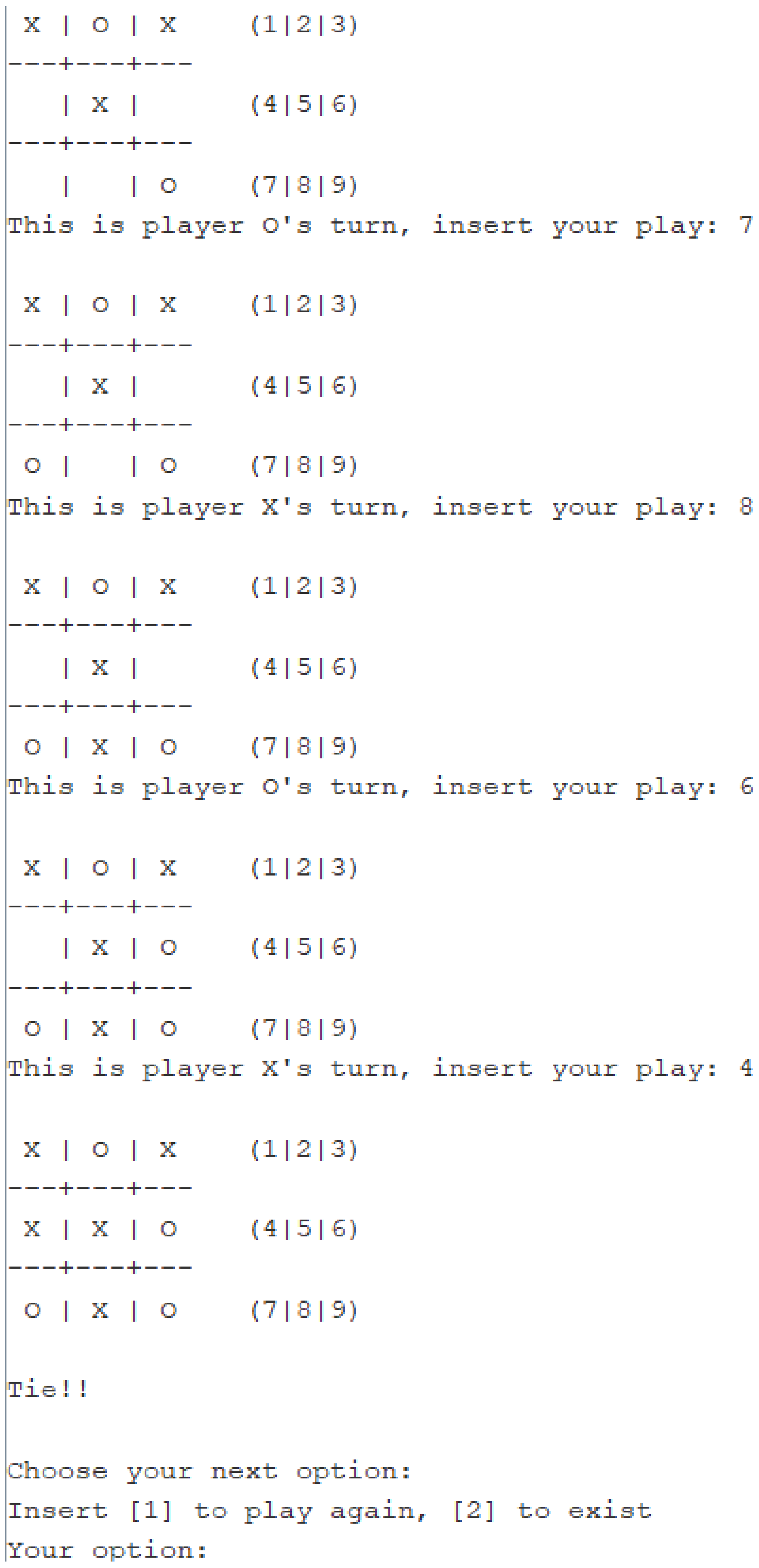
Example 1: Player O won!

Gameplay Example



```
   |   |        (1|2|3)
---+---+---
   |   |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player X's turn, insert your play: 1

 X |   |        (1|2|3)
---+---+---
   |   |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player O's turn, insert your play: 2

 X | O |        (1|2|3)
---+---+---
   |   |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player X's turn, insert your play: 5

 X | O |        (1|2|3)
---+---+---
   | X |        (4|5|6)
---+---+---
   |   |        (7|8|9)
This is player O's turn, insert your play: 9

 X | O |        (1|2|3)
---+---+---
   | X |        (4|5|6)
---+---+---
   |   | O      (7|8|9)
This is player X's turn, insert your play: 3
```

Gameplay Example



```
 X | O | X     (1|2|3)
---+---+---
   | X |       (4|5|6)
---+---+---
   |   | O     (7|8|9)
This is player O's turn, insert your play: 7

 X | O | X     (1|2|3)
---+---+---
   | X |       (4|5|6)
---+---+---
 O |   | O     (7|8|9)
This is player X's turn, insert your play: 8

 X | O | X     (1|2|3)
---+---+---
   | X |       (4|5|6)
---+---+---
 O | X | O     (7|8|9)
This is player O's turn, insert your play: 6

 X | O | X     (1|2|3)
---+---+---
   | X | O     (4|5|6)
---+---+---
 O | X | O     (7|8|9)
This is player X's turn, insert your play: 4

 X | O | X     (1|2|3)
---+---+---
 X | X | O     (4|5|6)
---+---+---
 O | X | O     (7|8|9)

Tie!!

Choose your next option:
Insert [1] to play again, [2] to exist
Your option:
```

Example 2: Tie!