

**COAL PROJECT REPORT**

**5 IN A ROW GAME**

**Group Members:**  
Laiba Mohsin - 22K-4246

Aniqa Azhar - 22K-4228

Aisha Jalil - 22K-4649

**Lab Instructor** : Kariz Kamal

**Section** : BCS -3K

5 IN A ROW GAME

**Introduction:**

The provided assembly language code presents an implementation of a 5 in a Row game, utilizing x86 architecture and the Irvine32 library. The game offers a turn-based experience where two players place their symbols (X or O) on a 5x5 game board. Accompanying features include a menu system, score tracking, a pause menu, and a basic login system. This project delves into the realm of low-level programming, demonstrating the use of assembly language for game development.

**Why We Chose This Project:**

The decision to undertake this project stems from a desire to explore the challenges and possibilities offered by programming games in assembly language. Additionally, game development allows us to combine logic, user interaction, and visual elements, offering a holistic programming experience.

Choosing a simplified version of the classic 5 in a Row game ensures that the project remains manageable while still presenting challenges in terms of game logic, user interface, and file handling. Moreover, implementing features like a pause menu and login system adds complexity, making the project more engaging and reflective of real-world applications.

**How the Game Works:**

The "5 in a Row" game is a strategic board game that typically involves two players who take turns marking a cell in a grid. The objective is to be the first to achieve a continuous sequence of five of their own symbols (usually X or O) in a row, either horizontally, vertically, or diagonally on the game board.

**Code Breakdown:**

Data Section:The data section contains various constants, prompts, messages, and variables used in the program, such as the game board, player scores, prompts for input, and messages for different game states.

Procedures: The code is organized into different procedures, each serving a specific purpose. Procedures like Menu, Game, BuildBoard, PrintBoard, CheckifEmpty, GetPiece, Read\_Coord, PauseMenu, and others are defined to handle different aspects of the game.There's a login system implemented with procedures such as Login, readUser, readPass, and CompareStrings for checking the entered username and password.

Game Logic: The game follows a turn-based logic, where players take turns entering their moves.The main game loop is controlled by the Game procedure, which includes options to start the game, display the scoreboard, or exit the program.The game board is initialized, displayed, and updated based on player moves. Winning conditions are checked after each move.

Score Tracking: Player scores are tracked and displayed in the scoreboard.There are options to reset the scores or exit the game.

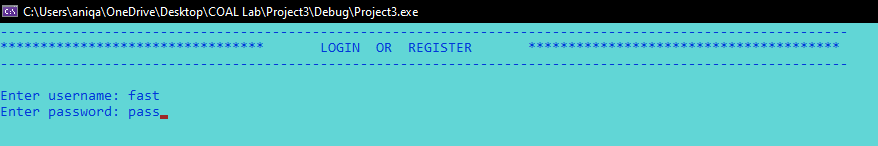
Menu and Pause Menu:The menu provides options to start the game, display the scoreboard, or exit.The pause menu allows players to reset the score, restart the game, continue, or quit.

Login System:Users are prompted to enter a username and password.The entered credentials are compared with predefined values (correct\_username and correct\_password).

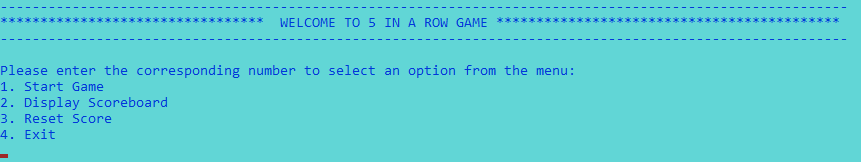
File Handling: File handling is used for reading usernames and passwords from files (username.txt and password.txt).If the files do not exist, the program allows users to sign up by creating these files.

**The Game:**

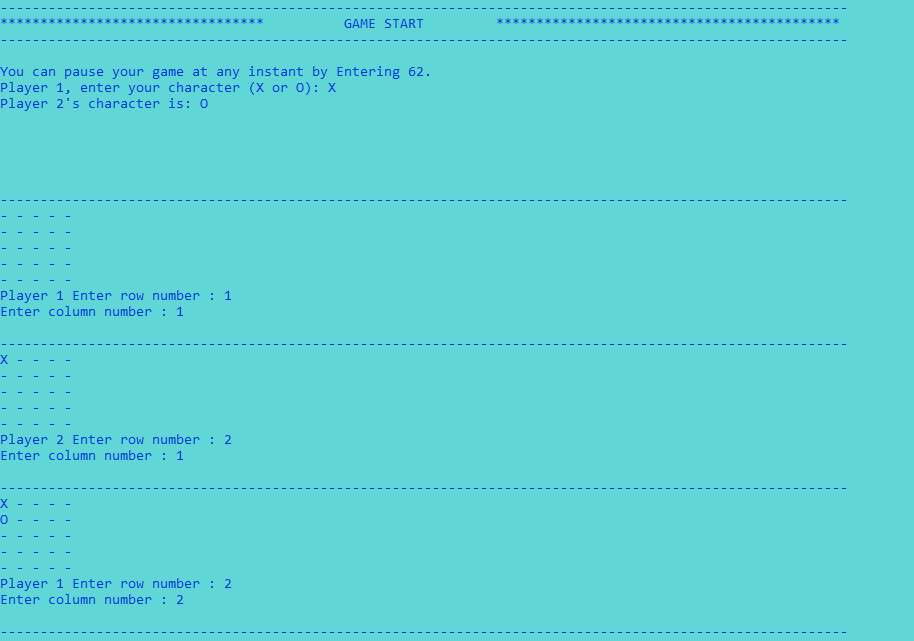
Login screen:

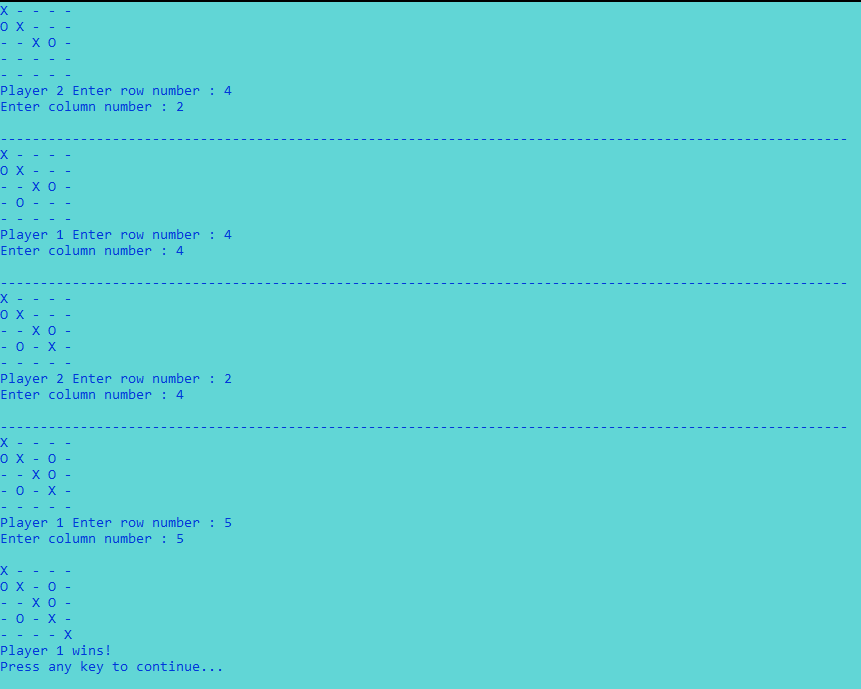
****

Menu:

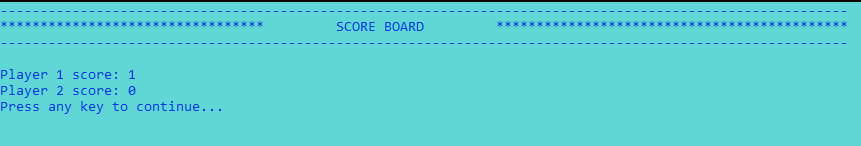
****

Option 1: Start Game

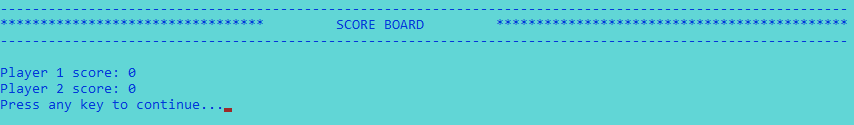
****

****

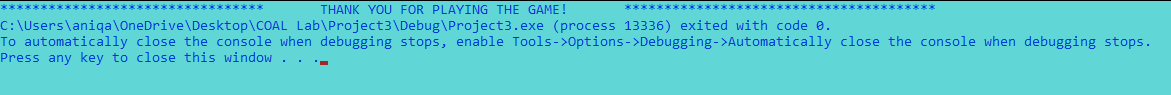
Option 2: Display ScoreBoard

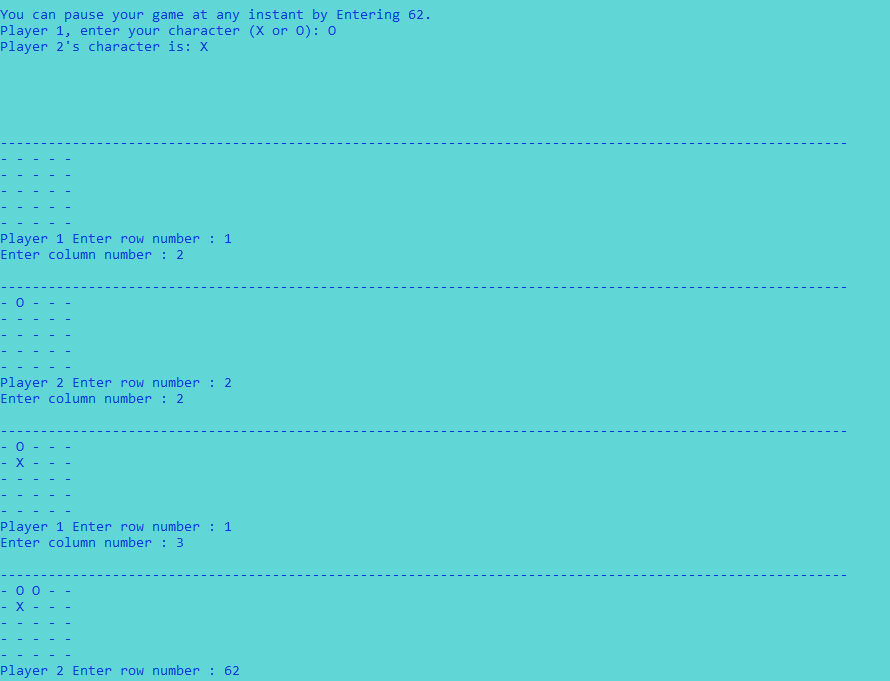


Option 3: Reset Score



Option 4: Exit



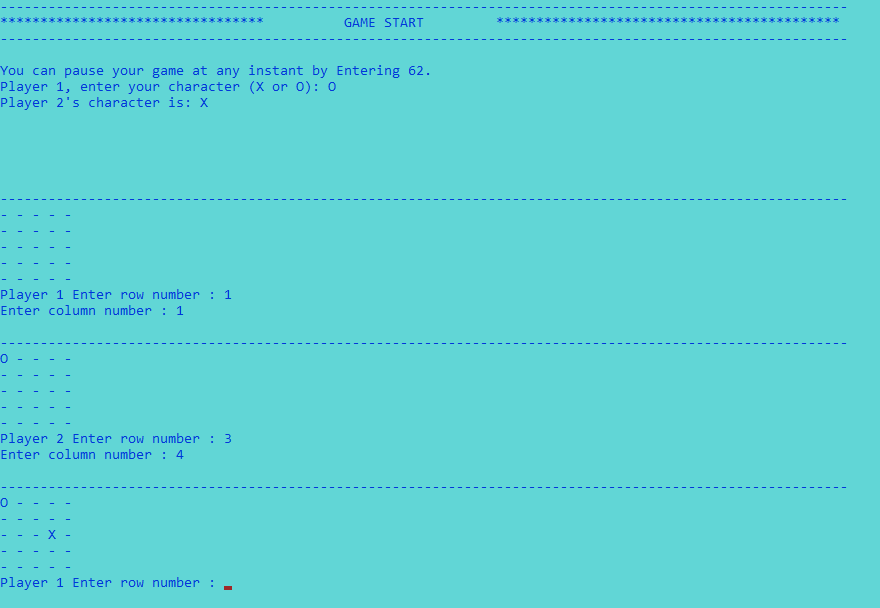
Pause menu:  




Pause Menu Option 4: Continue game



Pause menu Option 2: Restart Game



**Member Contributions:**  
Aniqa - Game PROC, BuildBoard PROC, PrintBoard PROC, checkifEmpty PROC, Menu PROC, GetPiece PROC, scoreboard PROC

Aisha - checkMatch PROC, RowCheck PROC, ColumnCheck PROC, DiagonalCheck PROC, read\_coord PROC, pause PROC, updateBoard PROC

Laiba - Winner PROC, Login PROC, FileExists PROC, SignorLog PROC, CreateFile PROC, CompareStrings PROC, ReadUser PROC, ReadPass PROC

Before beginning the implementation of the program, we collectively discussed and agreed upon the game's flow, logic, and overall structure. Each team member, as outlined in the "Member Contributions" section, took on specific responsibilities, and as issues arose during individual tasks, we collaborated to troubleshoot and find solutions.

**Conclusion:**

In conclusion, this project not only offers a glimpse into the intricacies of game development in assembly language but also showcases the adaptability of low-level programming for creating engaging and interactive applications. The chosen 5 in a Row game provides a balanced platform for exploration, learning, and the application of assembly language concepts.