**Hangman Game Report**

**Creators:** Syed Uzair Hussain(22k-4212) & Mussadiq Kamal(22k-4432)

* **Introduction**

The Hangman game is a classic word-guessing game where a player tries to deduce a hidden word by guessing individual letters. This report outlines the development, functionality, and features of a Hangman game implemented using Assembly language in Irvine library.

* **Problem Statement**

The objective was to create an interactive Hangman game in Assembly that allows users to guess letters within a word while visualizing the progress and maintaining a limit on incorrect guesses.

* **Motivation**

The motivation behind creating this Hangman game was to understand and apply fundamental programming concepts in Assembly language, such as string manipulation, conditional statements, loops, and user input/output handling.

* **Proposed Solution**

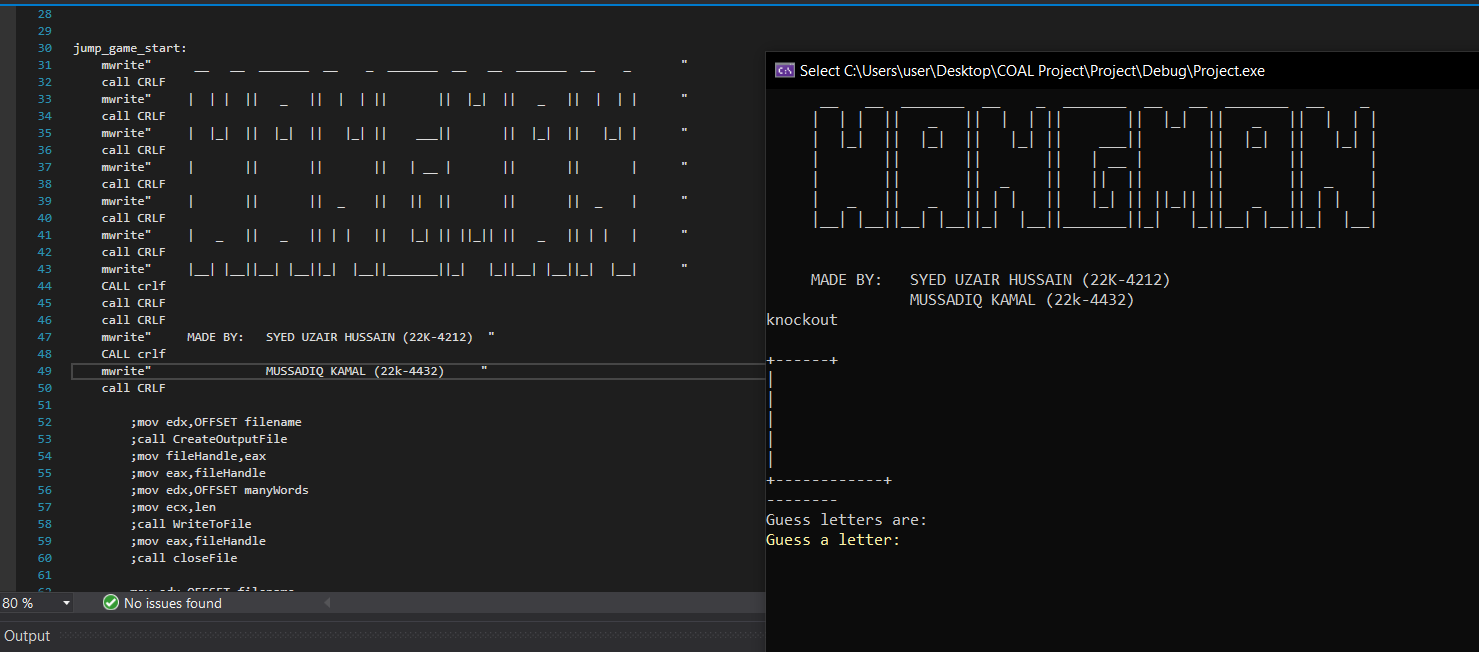
The Hangman game was implemented using Assembly language in Irvine library. The game generates a random word from a provided word list and prompts the user to guess letters. It displays the Hangman's gallows, with different stages representing incorrect guesses, and reveals correctly guessed letters within the hidden word.

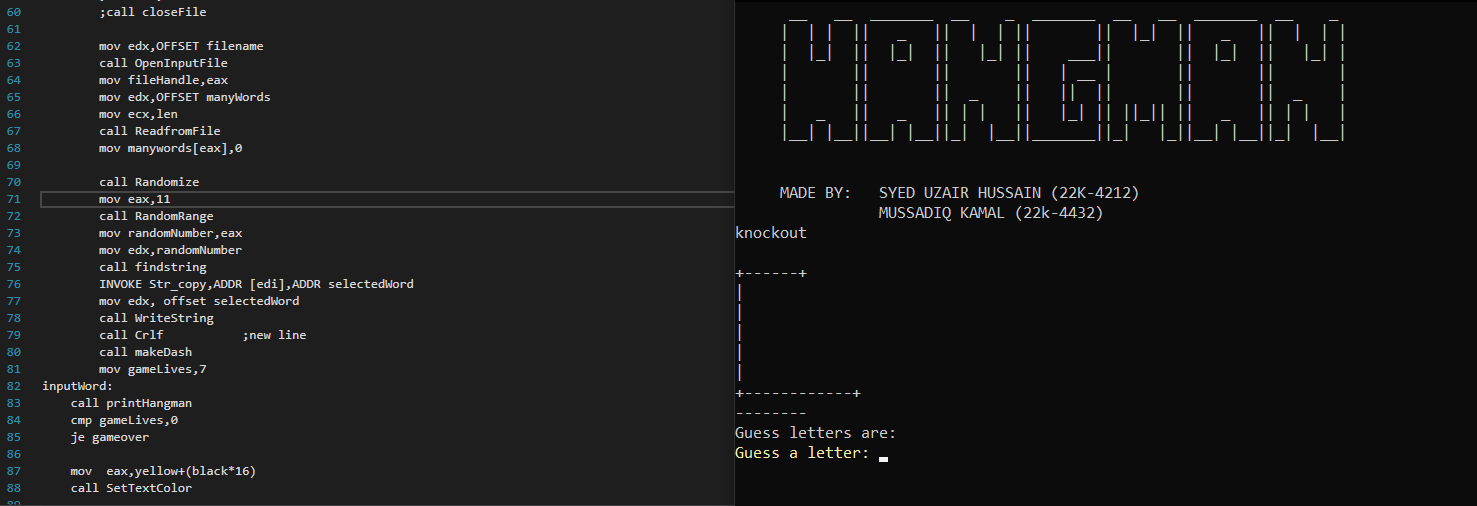
* **Methodology**

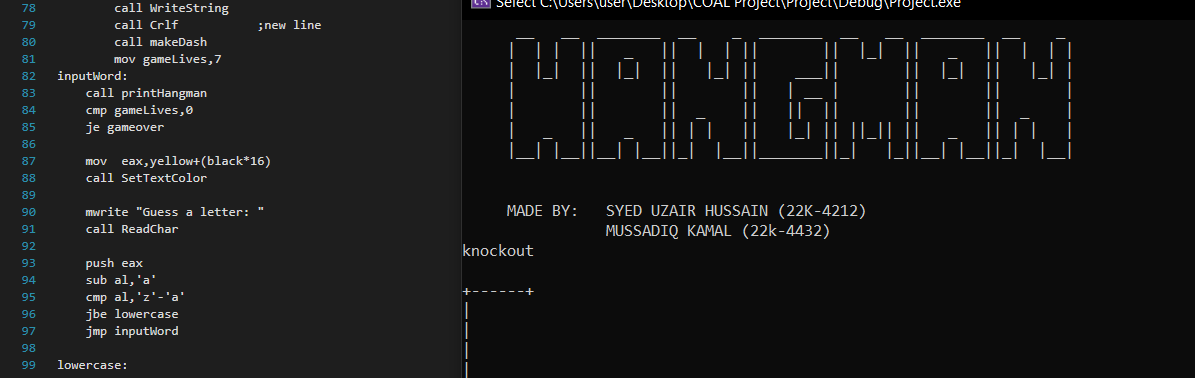
*-User Interface*: ASCII art is used for visual representation of the Hangman's stages of life.

*- Game Logic:* The game logic involves comparing user guesses with the selected word’s letters (bytes), and putting that word in guessWord array(initialized with dashes).If the guessed letters already has been guessed then the user is prompted to enter another letter.This goes in until the whole word has been guessed or the given number of lives have been expired. Two jump statements handle the gameWon logic and gameover logic and displays the progress to users simultaneously.

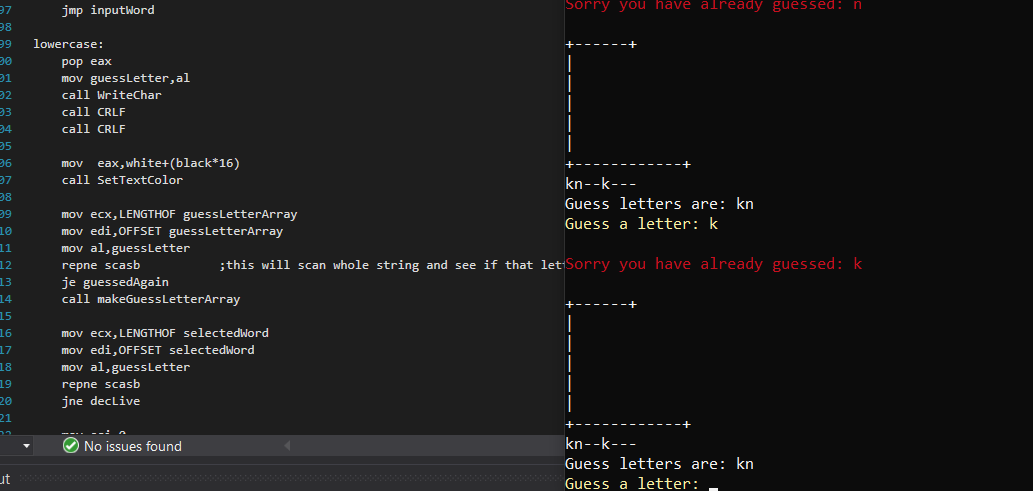
*- Input/Output Handling:* The game receives user input for letter guesses and provides visual feedback on correct and incorrect guesses of the word randomly selected from file (words.txt).

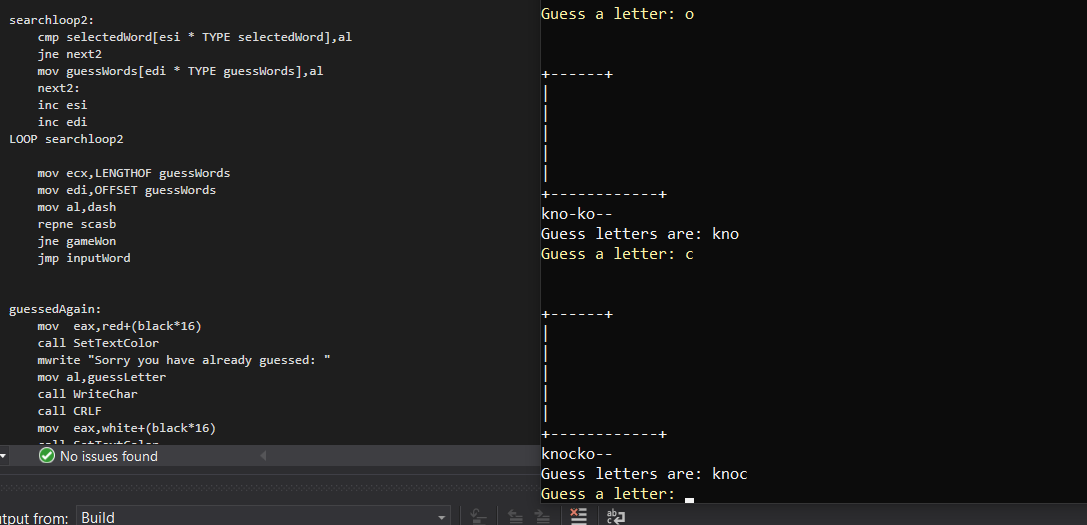


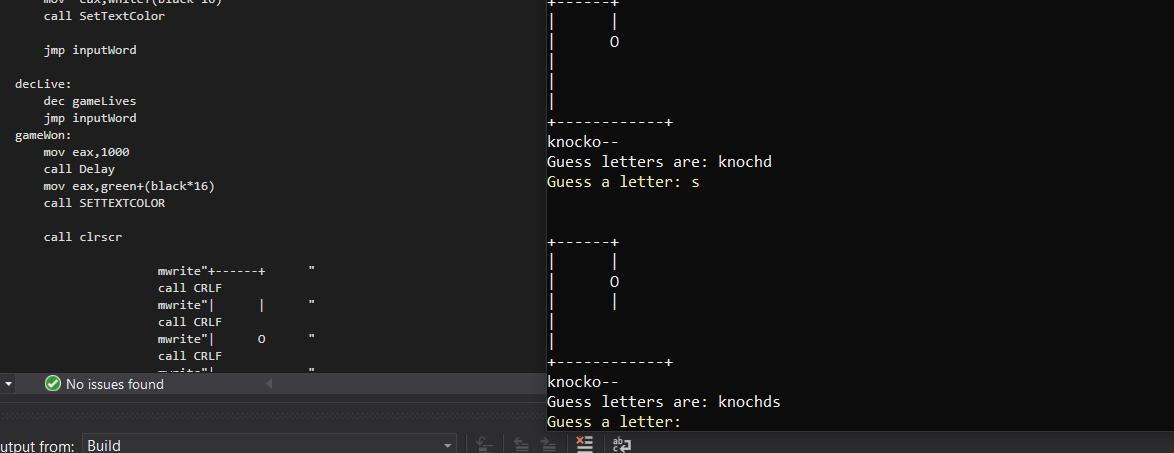


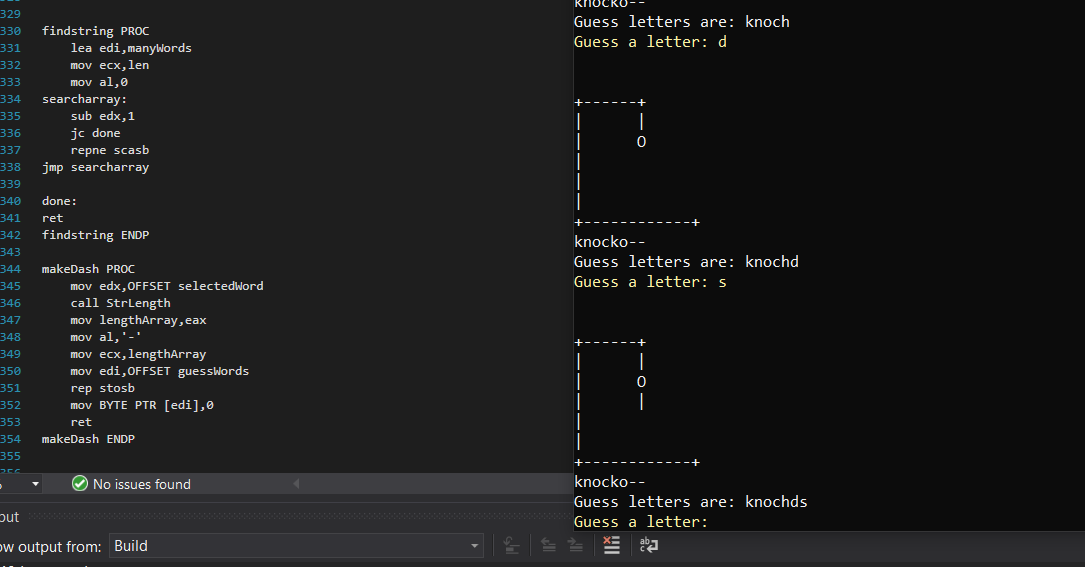


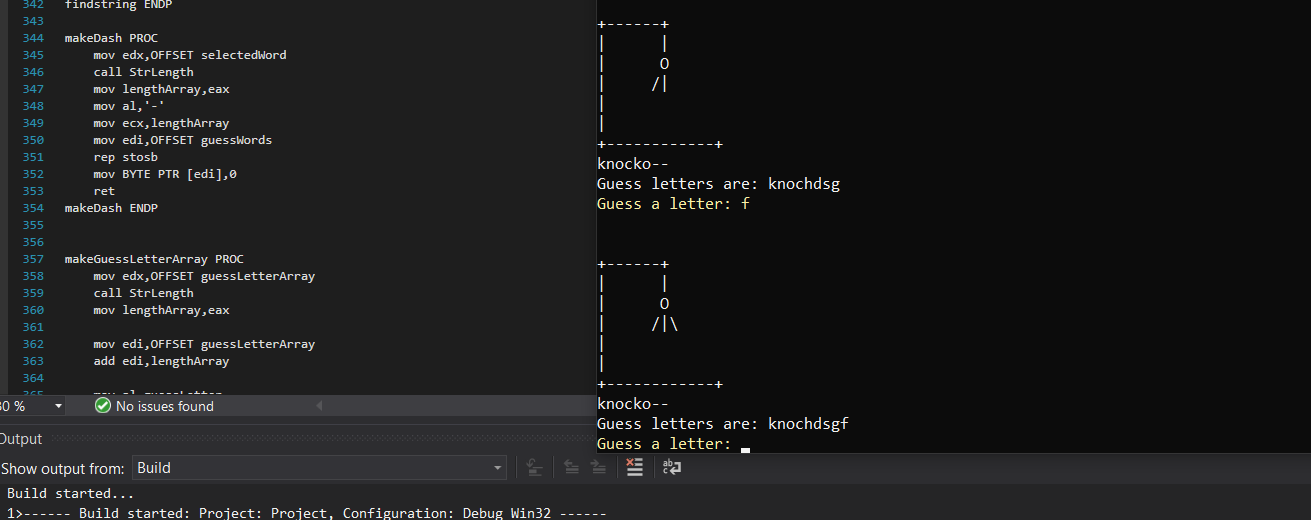


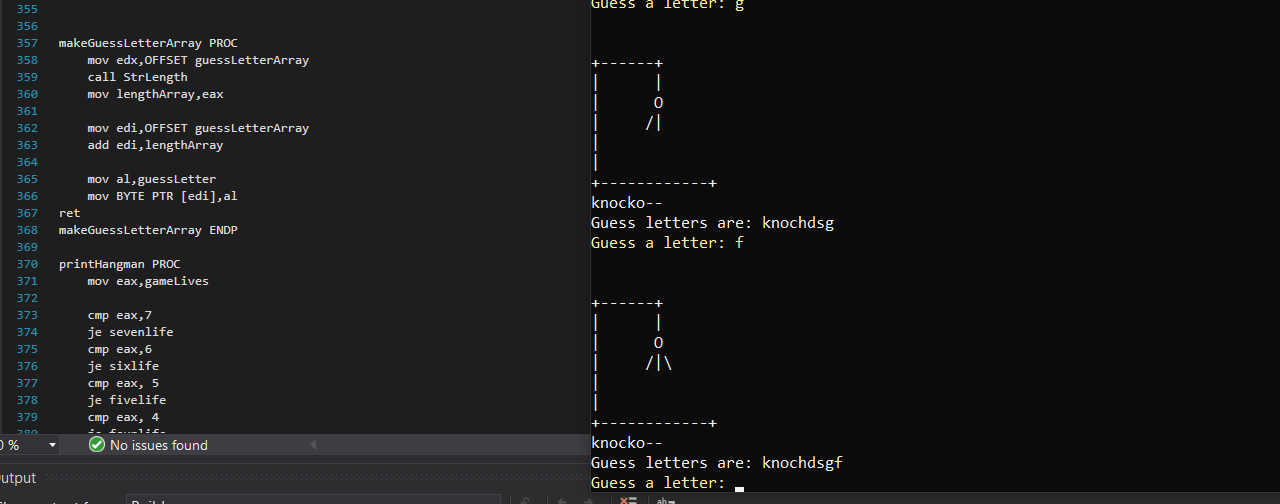
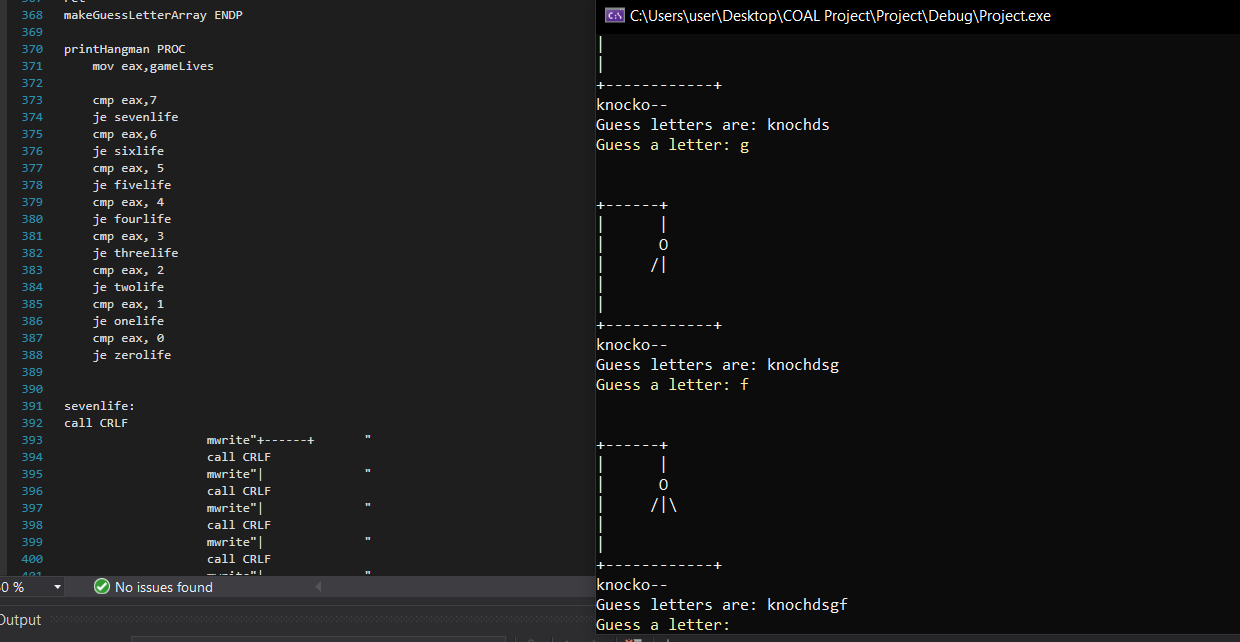
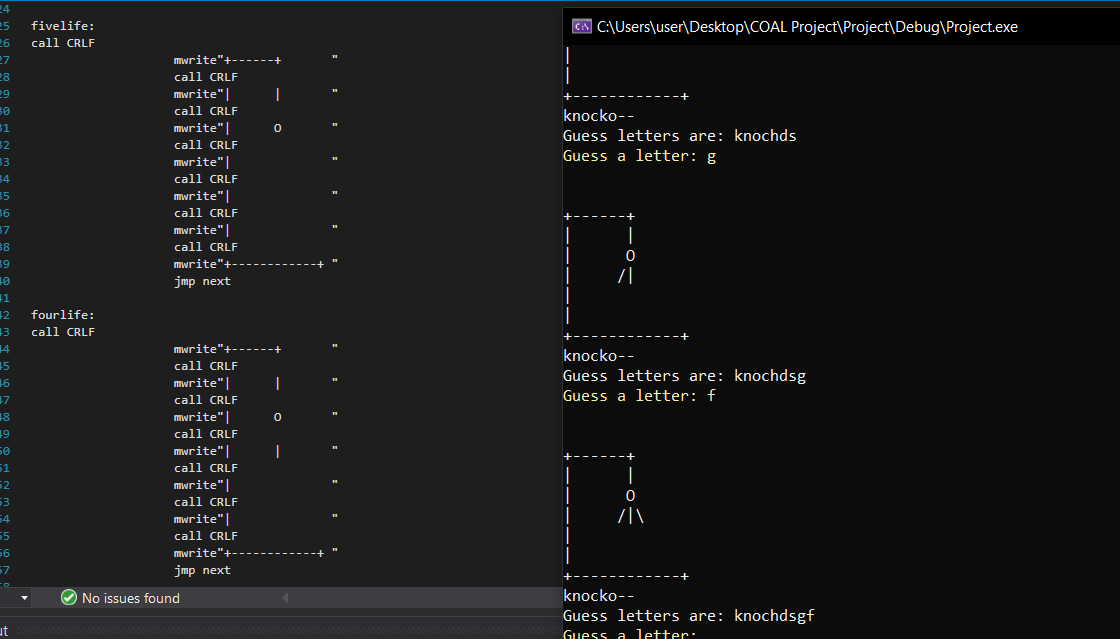
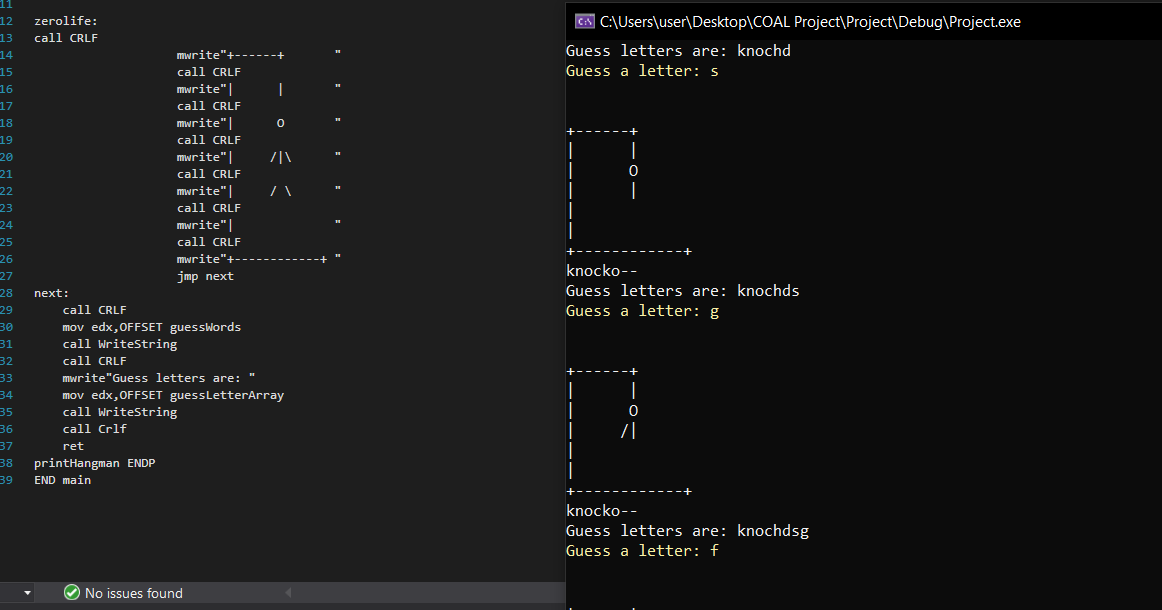
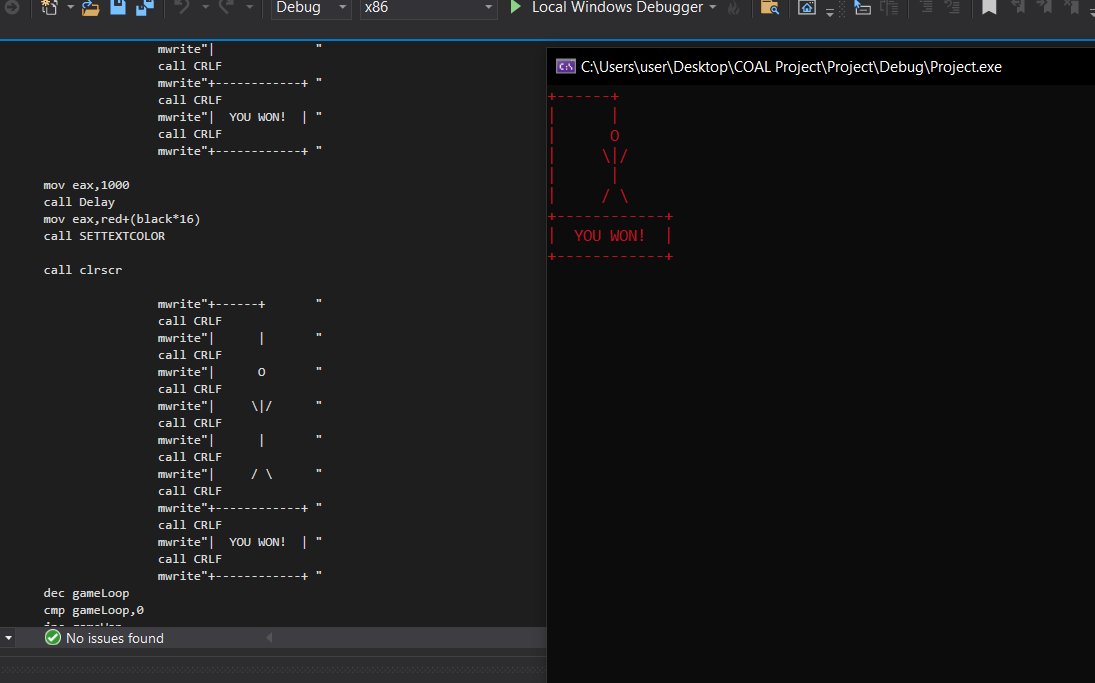


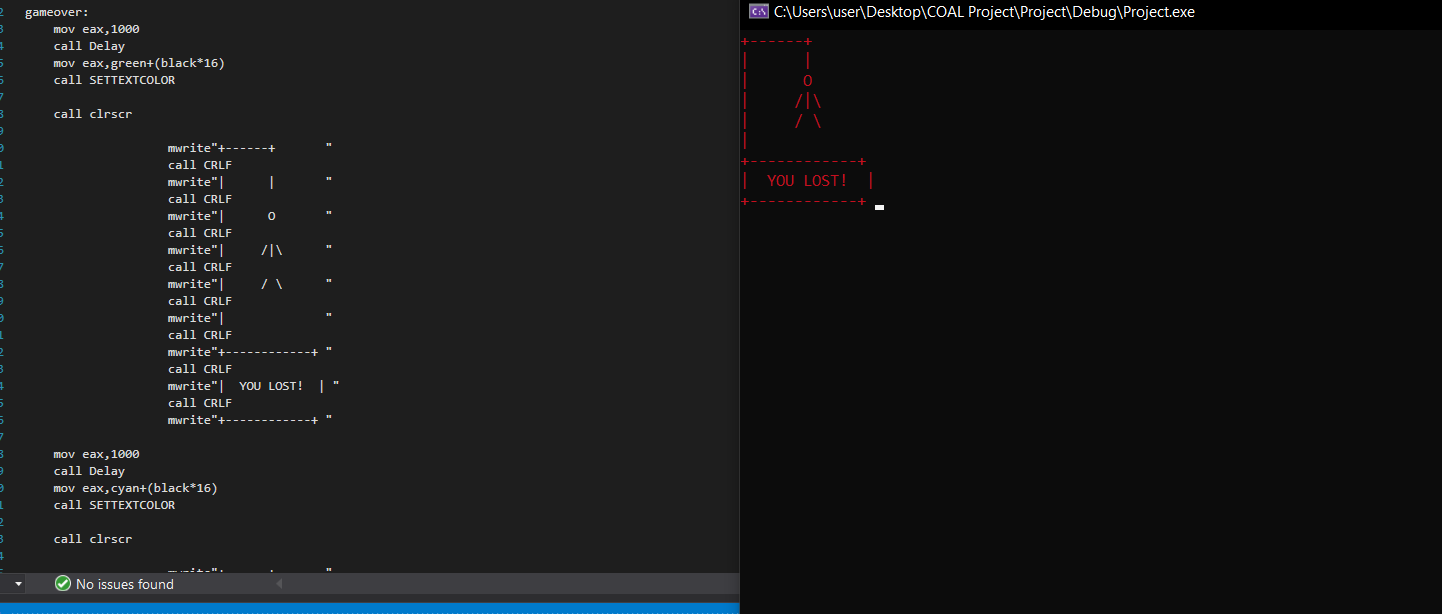










* **Results Explanation (VS Code Implementation)**

The code utilizes Irvine library and Macros library procedures for input/output, string manipulation, and file handling. It takes a random word from file, prompts the user for letter guesses, updates the Hangman visualization, and concludes when the user either guesses the word or exceeds the maximum allowed incorrect guesses.

* **Advantages/Disadvantages:**

*- Advantages:*

- Reinforcement of Assembly programming concepts taught in lab and learnt externally.

- Interactive and engaging gameplay experience.

*- Disadvantages:*

- Limited graphical capabilities in Assembly might restrict visual enhancements as well as inconvenience in coding due to limited readability (such as keeping track of jump statements,variables etc.)

* **Conclusion:**

The Hangman game implemented in Assembly language demonstrates the utilization of core programming principles to create an interactive gaming experience. This project provides valuable insights into low-level programming, game development and memory management of a machine. It was indeed an insightful experience to make this project.