**National University of Computer & Emerging Sciences**

**Karachi Campus**



**Project Report**

**Computer Organization & Assembly Language**

**Section:**

**Group Members:**

**PROJECT TITLE:**

**“WORD HUNT”**

* **Introduction**

In this project we have designed a game that challenges our user to find some hidden words in a table. The player is allowed to move in all direction from its current position in order to find a word correctly. The users are allowed as many attempts as they want until they give up.

* **Literature review:**
* *Back tracking:*

We will be using back tracking to trace out some words which we have in a dictionary array.

* *Use of 2D array:*

Since this project is sort of a word puzzle, we’ll using 2D arrays to represent the whole word grid and will be performing functions accordingly.

* *CMP:*

There will be an extensive use of compare instructions as we will have to compare multiple words in all 4 directions and backtrack accordingly.

* **Problem Definition:**
* The problem that accrue was that some users are unable to find the hidden words hence making the game difficult and results users’ interest.
* **Methodology:**
* we needed to find the words character by character, so the most efficient way to find the word was to apply a recursive approach that includes backtracking and also identify the words that were not found by the player.
* **Detailed Design & Architecture:**

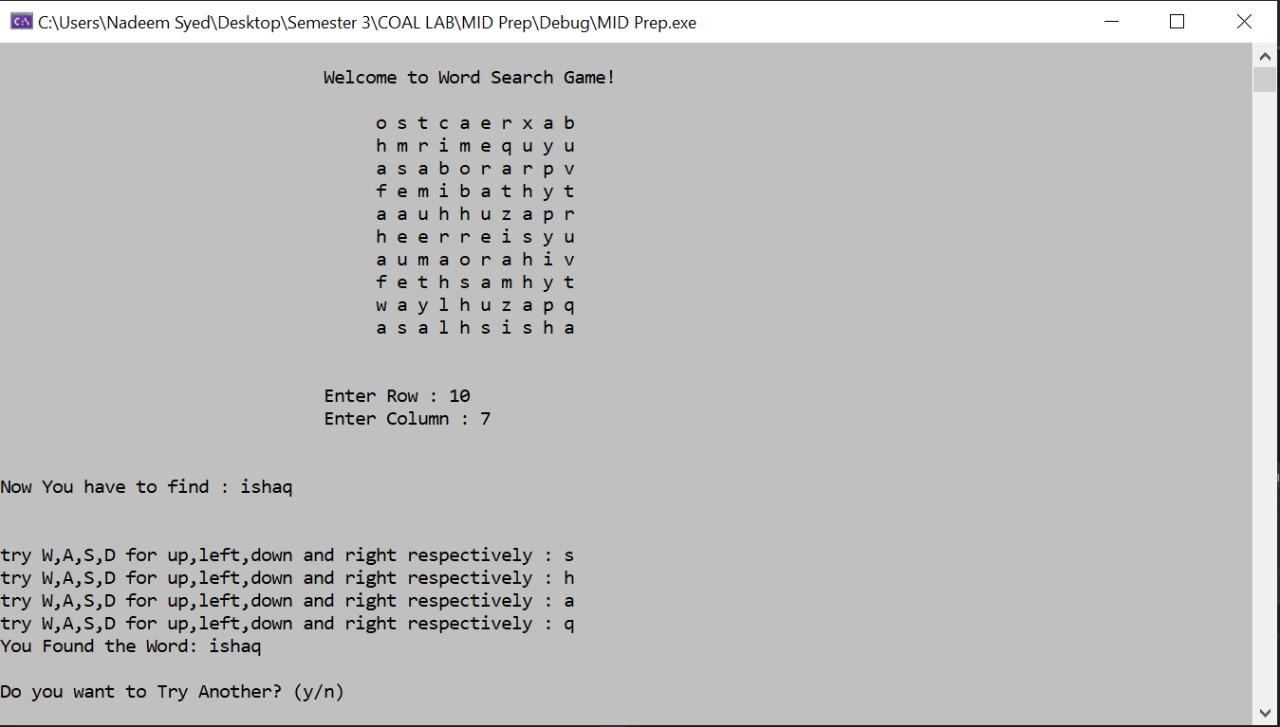
In this project we will be using multiple 2D arrays simultaneously and so our architecture will revolve around initializing these arrays resetting these arrays and printing these arrays. With the help of these arrays we’ll be performing searches in our original 2D array of word grid in which we will be searching all 4 directions. In the end when the user want to quit, we’ll be showing the solution of our word puzzle in different colors and this will be done by the use of backtracking.

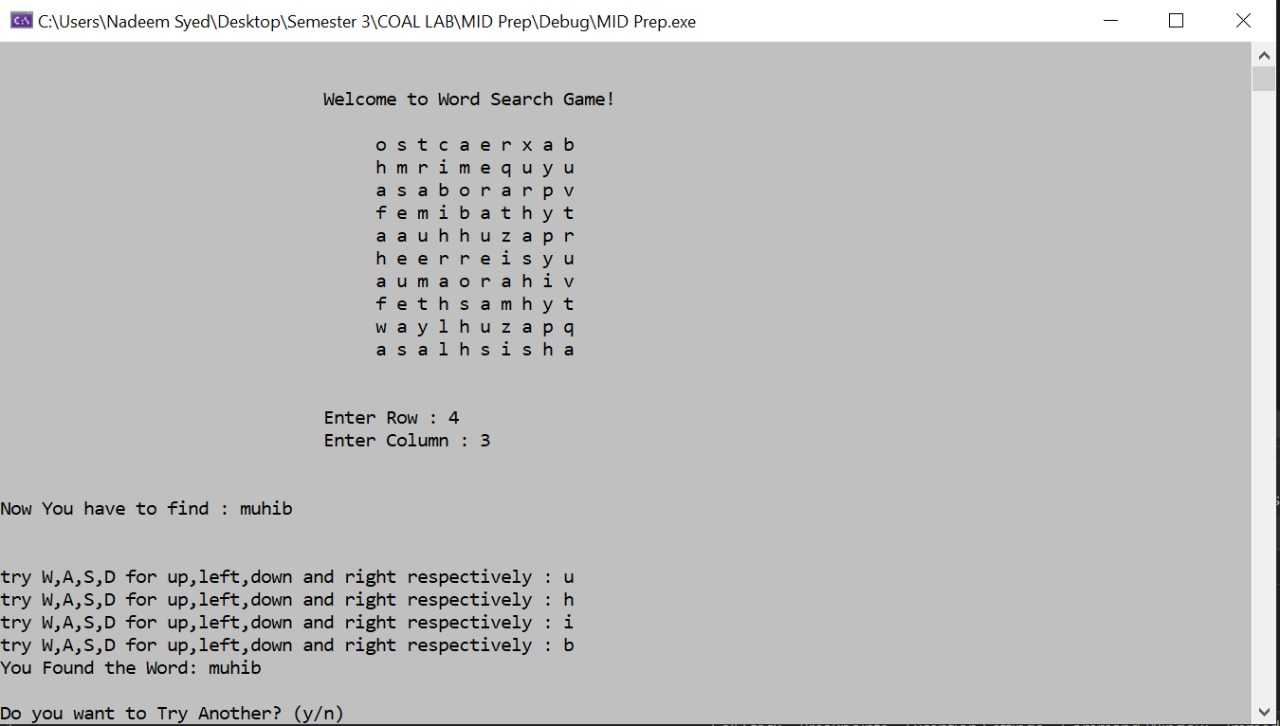
* **Implementation, testing and Programming code:**

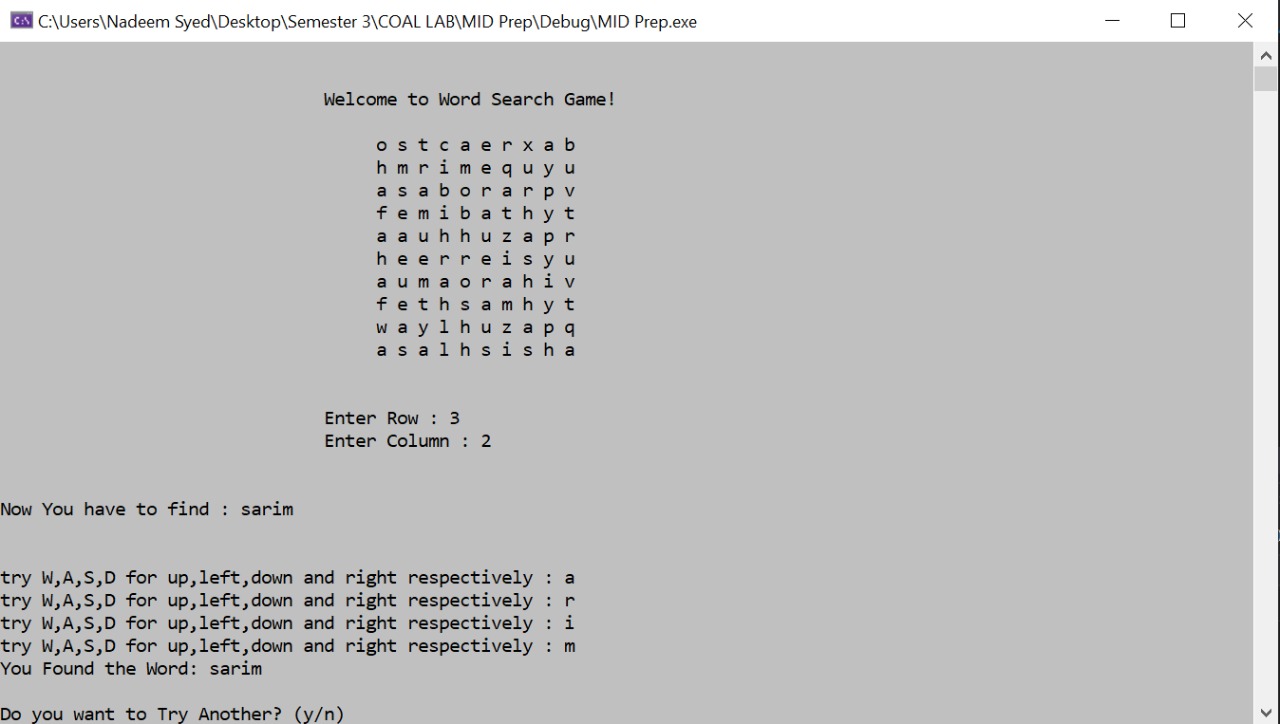
We will be implementing and testing the above described functions using the supportive environment of x86 assembly language and **Irvine** library.

* **Results/software simulation and discussion:**

**Test Cases:**

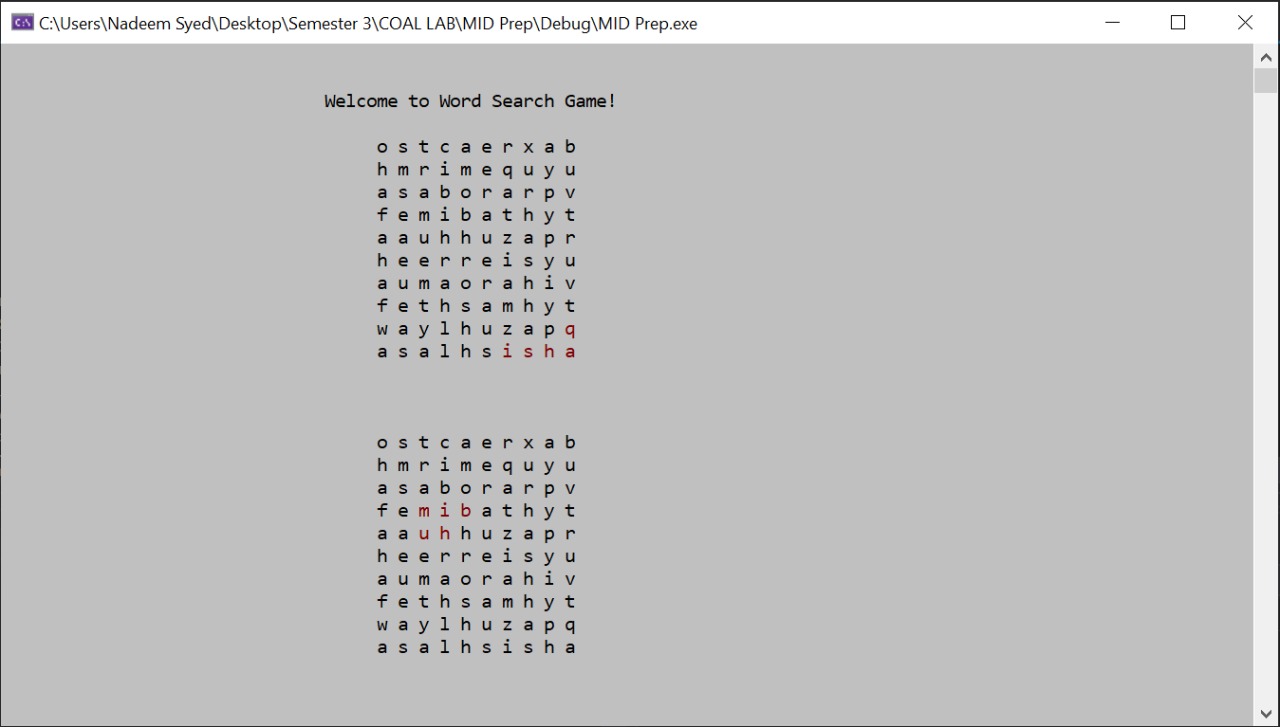


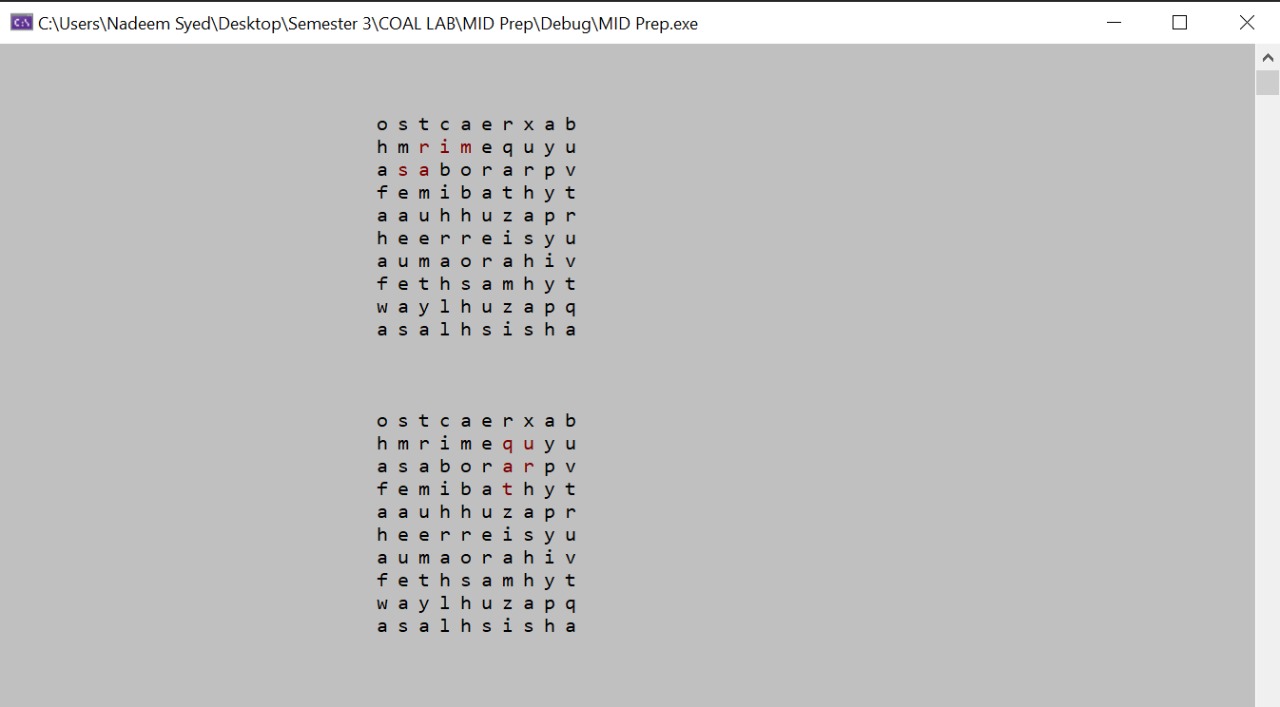




**Solution:**

When the user wants to quit the game all possible word (solution of the word puzzle) is highlighted in a different color.





* **Conclusion and Future Work:**

This project can be used as a game or a part of a game and is highly user interactive as the user can easily trace words using the basic game keys i.e W A S D.

* **References:**