**Project 2 Report**

# Youssef Samwel

## [yo800238@ucf.edu](mailto:yo800238@ucf.edu)

EEL3801: Computer Organization

Due Date: 7/23/2023

# **Project Description**

The goal of this project is to introduce the students to the algorithm development through string processing. String processing is one of the most common applications of algorithm and has wide range of applications. There are many applications of string processing but here is a brief list, search indexing, text documents, RegEx operations, Unicode, HTTP protocol, terminal, etc. Project 2’s technical objective was to count certain characters in a string that were determined by the following character set “KNIGHTS” (case insensitive). The students were first tasked to write a program using MIP’s assembly to accomplish the previously mention objective. Afterwards, the students were tasked to reduce energy consumption and dynamic instruction count (execution time) of their program.

# **2.0 Program Design**

To simplify the testing process, this program has the input string inside the data section of memory instead of taking user’s input.

## Part A

Execution began at the start label where we load the string address into $t0. Afterwards, we enter the “text loop” were we step through every character until NULL termination. For each character we invoke the “count letter” subroutine in which we process the letter. The letter must be stored in $t1 for the subroutine. The “count letter” subroutine will first convert the letter to lowercase and the load the character set “KNIGHTS” memory address into $s0. We step through each character in the character set and compare the current letter to the letter in the character set; if the current letter matches any letter in the character set, then we invoke the “valid charset letter” subroutine where we increment the count of specific letter in the counter integer array. We repeat this process until the end of the string. Afterwards, we invoke print\_output with $t7 set to 1 to display the count of each letter in the console output and we call print\_output again with $t7 set to 0 to display the count of each letter using a histogram format in the console output.

## Part B

Part B also achieves the same output as Part A but with reduced dynamic instruction count and reduced energy consumption. Execution also begins at the start label where we load the input string into $t7 and step through each character until NULL termination.

A diagram of a computer program

Description automatically generated

Part A

# **3.0 Symbol Table**

# **4.0 Learning Coverage**

# **5.0 Prototype in C-Language**

# **6.0 Test Plan**

# **7.0 Test Results**

# **8.0 References**