CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

Introduction

1.1 Purpose

The purpose of this program is to count the letter inputs of a person's name.

1.2 Intended Audience and Users

The primary audience/user is any individual interested in counting how many letters are in their name or any LC-3 Assembly enthusiast.

1.3 Product Scope

The intention of this program is to first prompt the user to type their name, one letter at a time. Then the program will take in the given information and then continue to loop the input until the user presses the 'ENTER' key. After this step, the program will store the given information and then it will output the given information.

1.4 Reference

Source Documents for the Program Requirements and Specification

(Lc3 manual)

(Final Exam Recommendation Draft Document)
(https://www.cs.colostate.edu/~fsieker/misc/CtoLC3.html)

Companion Application Requirements Documents (If applicable)

Not Applicable

2. Overall Description

2.1 Product Perspective

The CIS provides:

A service to count the characters input by the user

- including alphabet
- excluding numbers
- excluding special characters

2.2 Product Functions

The overall description of functionality:

1. Provides a direct and easy to use interface for the user
2. Contains two looping subtasks for the input and output of user's input data
3. Outputs alpha-numerical results corresponding to the user input

Technical functionality

A configurable toolkit of functions including:

- Ability to define the alphabetical input of the user
- Ability to configure the alphabetical input and convert it to an ASCII equivalent
- Ability to store the given data, count it, and execute by form of an output

2.3 User Classes and Characteristics

Student

I am the only developer involved in the creation of this program.

2.4 Operating Environment

This type of application will be run using the LC-3 Assembly editor and simulator on any Windows OS.

2.5 Design and Implementation Constraints

This application requires the LC-3 Assembly program to run. The program also outputs in an ASCII format, so if an input of 10 or more of the same letter are inserted, the output will be ASCII equivalent past the number 9.

2.6 Assumptions and Dependencies

It is assumed that the user is competent in using a keyboard and has access to one.

The program will depend in using the proper software, in order for it to run.

3. External Interface Requirements

3.1 User Interfaces

The user has to type in their name. The user can input data with the help of the keyboard. After they have completed, they will press the enter key on their keyboard and the program will output the inserted data from the user, counting the amount of characters that were inputted in the program.

3.2 Hardware Interfaces

The user will need to use a PC computer or laptop to run the software.

3.3 Software Interfaces

The LC-3 Assembly editor and simulator are required to operate the code to run the application.

3.4 Communications Interface

This application does not require any internet connect and can be ran offline.

4. Detailed Description of Functional requirements

4.1 Type of Requirement (summarize from Section 2.2)

Purpose: Provides the user with a counter for the letters they input

Inputs: Inputs are through the keyboard.

Processing: The input is looped and stored until the user hits the enter key.

Outputs: The alphabet is output with the number of times they were input by outputting an ASCII numerical value.

Data: User database

4.2 Performance requirements

4.2.1 The application should be portable and possible to hold letters contained in large names that are input by the users

4.2.2 The response time for input should be almost immediate.

4.3 Flowchart and Pseudocode.

Pseudocode:

Prompt: Enter your name a letter at a time and press enter when done

Process:

1. User inputs variable
2. If any key but the ENTER key is pressed, the loop will continue to ask for user input.
3. If ENTER key is pressed, display output
4. The stored letters are converted to ASCII decimal value to count how many letters were input and outputs the ASCII number value from 0 to 9 with letter A to Z

5. Program ends

Flowchart: