CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

Introduction(Lucas)

1.1 Purpose

The purpose of the project is to test our ability to collaborate as a team to fully create every part of an LC-3 program from the documentation to the LC-3 code.

1.2 Intended Audience and Users

This program has a general demographic

1.3 Product Scope

What is the intention of this program?
This program will display the frequency of a character in the full name that the user enters.

1.4 Reference

Source Documents for the Program Requirements and Specification

Reference Project requirements and LC-3 specifications.

Example

1) Curricular Information System Vision and Scope
   http://web.mit.edu/ssit/cis/CISVisionScope.html
2) HTML/Print Generator and Document Management Sub-Project Vision and Scope
3) CIS Project Participants
   http://web.mit.edu/ssit/cis/CISpeople.html
4) Catalog Production Schedule Timeline
5) CSB and Registration Timeline
6) Old Catalog Data description (Clipper system, web proposal system, MITSIS)
7) Scheduling system Class Schedule Book (CSB) File layout and description
8) Old Catalog Functions (spread sheet)
9) CSB and Registration Program documentation
10) Formal Department and Academic Services Interview Documents (detailed list available)
11) Final Exam Recommendation Draft Document

Companion Application Requirements Documents (If applicable)

What other documents should be reviewed with this document?

1) readme.txt

2. Overall Description

2.1 Product Perspective

The primary objectives consist of:
- Counting the frequency of letters within a name that is listed.
- Output a numerical value to represent the frequency of character in the users fullname.

2.2 Product Functions (Dylan)

The overall description of functionality:

Highlight the program functionality: Identify tasks and subtasks of the program in summary.

The program is created to simply count the characters within a full name. It allows a user to quickly receive the number of a certain character that is used within someone's name. This program is also created to provide an easy to use and intuitive experience when enabling a user to count the letters in their name.

Technical functionality

A configurable toolkit of functions including:
What are the technical functions of the program? Subroutines and operations.
The technical functions include:
- arithmetic, data movement and conditional operations.
- implement subroutines and their respective function calls.
- manage overflow and storage allocation.
- implement ASCII conversion operations.
- use conditional and iterative branching operations.

2.3 User Classes and Characteristics

Who are involved in this development process? Include business and technical personnel and their tasks.

The individuals who are involved:

Dylan Kral

Documentation

Lucas Wilkerson

Documentation

Alan Tran

Documentation

Manny Saucedo

Documentation

Example:

Academic Services personnel

Responsible for the overall tracking and publishing of the MIT catalog.
Support the development of new and changed subject proposals.
Support COC and CGSP review of new and changed proposals.
Pre-register and register students. Manage Add and Drop requests.
Schedule classrooms, students, and finals.
Manage and report on pre-requisites, co-requisites.
Audit student degrees (GIR)

Department Coordinators

Responsible for helping faculty develop MIT catalog and related information for their department.
Monitor departmental roadmaps
Help develop room schedules for subjects and exams
Audit department degrees

COC and CGSP

Review subject proposals

Other Administrative Offices

The HASS Office, PSB, Communications Office review and support the development of the MIT catalog and supplemental bulletins.
Run student lotteries.
Submit grades.

Faculty

Plan and teach curricula
Use many reports provided by Academic Services: class lists, etc.

Students

Use catalog and related information to plan course work.
Use the on-line planning worksheet, lottery submittal, and pre-registration functions.

2.4 Operating Environment

What type of system will the application be operated on? Operating system? System types? Development platform?
The system will be required to run on any 16 bit or higher computer. The operating system that is recommended is windows 10 with any of the current updates. New code would be developed on the LC-3 Editor.

2.5 Design and Implementation Constraints

Note any constraints or limitations to the application.

none.

Example: Access to the web is required. As for the developer constraints, the alumni information was not available for security reasons. Many assumptions about the data had to be made. There is high learning curve.

2.6 Assumptions and Dependencies

Note any dependencies

It is assumed that the user knows how to use their respective peripherals, such as mouse keyboard and monitor. Since the application is used on LC-3, no internet connection is required to use the program.

3. External Interface Requirements (Manny)

3.1 User Interfaces

How will the user interface with your program? Menus? Access prompt? Links? Icons?

The program user will begin by opening the LC-3 icon, which in turn will initialize the program. After assembling and loading the program into the simulator the user will utilize the keyboard to input the necessary data creating the desired output. Our program will be available through most popular platforms including MAC and PC and is compatible with most operating systems.

3.2 Hardware Interfaces

Specify hardware interface – computer types? Terminal types?

This program is a very simple one which at the very minimum requires a 16-bit system to fully operate. This minimum requirement will allow the program to be used in a vast array of computer systems with virtually no additional hardware required.

3.3 Software Interfaces

Specify additional software interface – if any. What type of software will the application require to run?

Aside from the LC-3 software used to execute this program, there are currently no other software requirements needed to successfully use our program. The emphasis was made to keep our program accessible to most users, this was achieved by keeping the necessary software to bare minimum.

3.4 Communications Interface

Does your application require web, Internet or network connectivity? If so, which browser? What type of network connection?

Currently our program does not require web, internet or network connectivity, if in the future our program transitions to a web-based platform, again the emphasis will be made for the program to remain as user friendly as possible.

4. Detailed Description of Functional requirements (Alan and Lucas)

4.1 Type of Requirement (summarize from Section 2.2)

What are the functions? Their purposes? Inputs? Outputs? Data? Where is the data stored (internal or external to the application)?

Program: Character Counter
Purpose: Displays the frequency of characters of user full name.

Inputs: Input is placed through keyboard.

Processing: The input is processed and verified through counting each character in the name of the user that is inputted.

Outputs: The output will be in numerical form dependent on the frequency of characters of the user's name. An error will appear if a character is entered that is not in the parameters of the data of the program.

Data: American Alphabet.

4.2 Performance requirements

What is the expected performance level of the program?

4.2.1 The application is expected to be able to perform on Windows 10 OS as well as without internet.

4.2.2 The application is only meant to count characters in a User's full name. Thus, the program should be able to analyze the characters in 1-2 seconds for the average name.

4.2.3 The database is the American Alphabet which consists of 26 unique characters. Based on how small the data.

4.2.4 Errors will be displayed if an a character is entered that is not registered in the database which consists of the American Alphabet.

4.3 Flow Chart and Pseudocode. (Lucas)

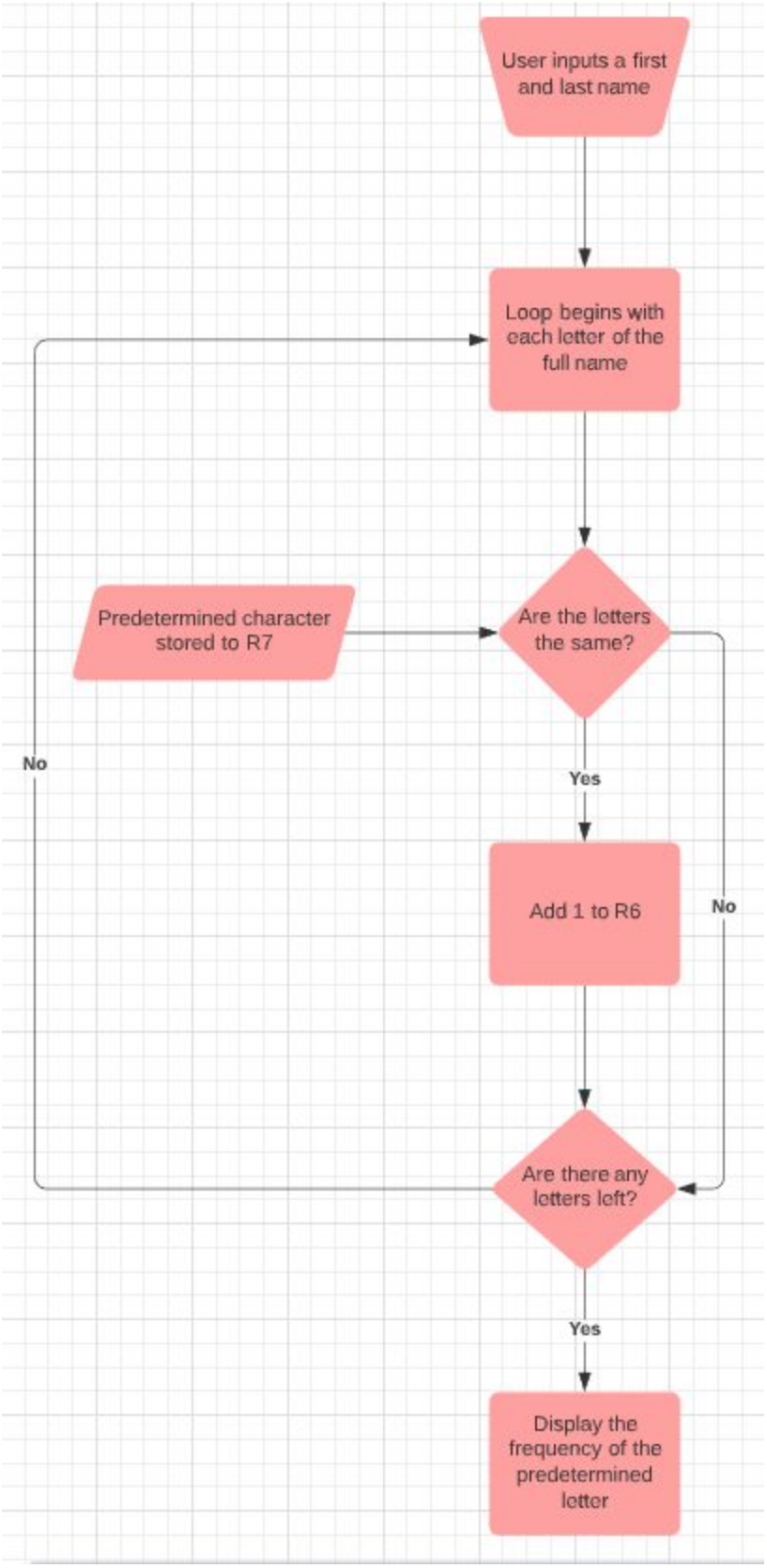
The program prints a line asking the user to enter their name.

The user is able to enter a first and last name.

The program should go through a loop that compares the chosen character to every letter in the name.

Every time the loop hits that letter it should add to R6.

After exiting the loop, display how many times that letter appeared.


User inputs a first and last name
Loop begins with each letter of the full name
Predetermined character stored to R7
Are the letters the same?
No
Yes
Add 1 to R6
No
Are there any letters left?
Yes
Display the frequency of the predetermined letter