CIS11 Course Project Part 1: Documenting the Project

Introduction

1. Purpose

The purpose of the Grading Calculator is to work with 5 numbers inputted by the user and to in return output the maximum, minimum, average score as well as the corresponding letter grade to the average.
Input: Decimal Numbers by user representing test scores
Output: Display on Console the maximum, minimum, average score and letter grade equivalence (0 - 50 = F, 60 - 69 = D, 70-79 = C, 80 - 89 = B, 90 - 100 = A)

2. Intended Audience and Users

- Specific Users: Students, Teachers, Education
- General Users

3. Product Scope

The intention of the Grading Calculator is to calculate the maximum, minimum and average scores of 5 given test scores by the user. It allows the user to compare test scores of their own or of a given 5 different students. The usefulness of the program is to gain an understanding of progress or growth of a test group.

4. Reference

Source Documents for the Program Requirements and Specification

1) LC-3 Editor
2) LC-3 Simulator

Companion Application Requirements Documents (If applicable)

1) LC-3 Editor
2) LC-3 Simulator
3) LC-3 Executable File

2. Overall Description

2.1 Product Perspective

The Grading Calculator provides:
A review on 5 varying test scores:
· Maximum score amongst the 5
- Sorting the 5 pieces of data from greatest to smallest
· Minimum score amongst the 5
- Sorting the 5 pieces of data from least to greatest

· Average score of all 5
  - Adding all 5 pieces of data and then dividing by 5 to get the average
· Letter grade complimenting average
  - A, B, C, D, F letter grade

All data outputted to the screen for the user of the program.

2. Product Functions

The overall description of functionality:

1. Prompt User for input
2. Check to see if input is valid- Is the user inputting a decimal value and negative values invalid.
3. Convert to ASCII - User input will be decimal values but LC-3 simulator runs on ASCII values so we will have to offset by 48 - 16 x3 offset
4. CALL JSR SUBROUTINES
5. MAX - Sort data to find the maximum value in the data, return to main calling program
6. MIN - Sort data to find the minimum value in the data, return to main calling program
7. AVERAGE - Add all 5 numbers and divide by 5
8. Use Average to find the equivalent Letter Grade
9. Convert back to Decimal Value
10. PUTS - Displaying values on console (MAX, MIN, AVERAGE, LETTER GRADE)
11. END program

Technical functionality

Functions are called through JSR subroutines (MAX, MIN, AVG, GRADE) that initialize the operation performance and once completed in the subroutine we return to the main calling program by JMP R7 - which is also known as RET. This saves the old PC and returns to the correct spot in the program by telling us how to get back to the current instruction.

3. User Classes and Characteristics

Developers

The Assembly Guys
Eric Perez
Tyler Danh
Shadoe Stiede
Valerie Avalos

Evaluator

Dr. Kasey Nguyen

Faculty User

Use many reports provided by calculator for glimpse of progress of class as a whole or individual students across time.

Students User

Use reports to track and predict progress in given course.

General User

Evaluate data of 5 different numerical data

4. Operating Environment

- Grading Calculator is developed for most efficient use on the Windows system
- Under any web server - Web server is not needed unless accessing simulator on a MAC in which case user will need to reach: https://wchargin.github.io/lc3web/ or another type of online simulator.
- Code developed and edited on the LC-3 Editor
- Program is tested on the LC-3 Simulator

5. Design and Implementation Constraints

- Numbers too large or too many entered that will cause overflow of the program if they are not moved around correctly in registers
- Numbers are not inputted correctly
- Machine is not reinitialized so that the program is at prime potential
- Maximum and Ideal is 5 pieces of data

6. Assumptions and Dependencies

- It is assumed that the user will input 5 values, the program will wait for all values to be entered. It is assumed that the user is familiar with using the simulator, reinitializing the machine after each run through, familiar with using the keyboard and mouse.

3. External Interface Requirements

1. User Interfaces

- User will input data after being prompted on console to enter the 5 test scores of their choice. They will input data using the keyboard numbers. Finally to enter their data, they can press enter. There are no navigation menus in the program.

2. Hardware Interfaces

PC computers - can run software and LC-3 editor/simulator
MAC Computers - must find an online editor/simulator

3. **Software Interfaces**

Best software to run on is Windows
Can be ran on another software if the user has access to internet connection to use an online simulator

4. **Communications Interface**

PC- Requires no internet connectivity while using LC-3 Simulator to test program
MAC - requires any type of internet connectivity to run an online LC-3 Simulator.

4. Detailed Description of Functional requirements

4.1 Type of Requirement (summarize from Section 2.2)

Purpose: Provides data on 5 test scores - max, min, average score and letter grade.

Inputs: Inputs are through the keyboard numbers and enter key.

Processing: The input is verified by checking if user entered a value then converted to ASCII value before performing subroutines and before outputting information to the user.

Outputs: The correct input will result in 4 pieces of information displayed on the console.
1) The maximum of 5 scores
2) The minimum of 5 scores
3) The average of 5 scores
4) The letter grade of the score

Data: 5 decimal numbers inputted by user.

4.2 Performance requirements

4.2.1 The application should be portable and possible to users of Internet Explorer.

4.2.2 The application will be displaying 3 values and a Letter Grade after its ran, the response time for this particular program should not be greater than 3-4 seconds after the user has entered all of their given data

4.2.3 The database should be scalable; it must have the capacity to hold 5 large num-

bers used for data but nothing outrageously large.

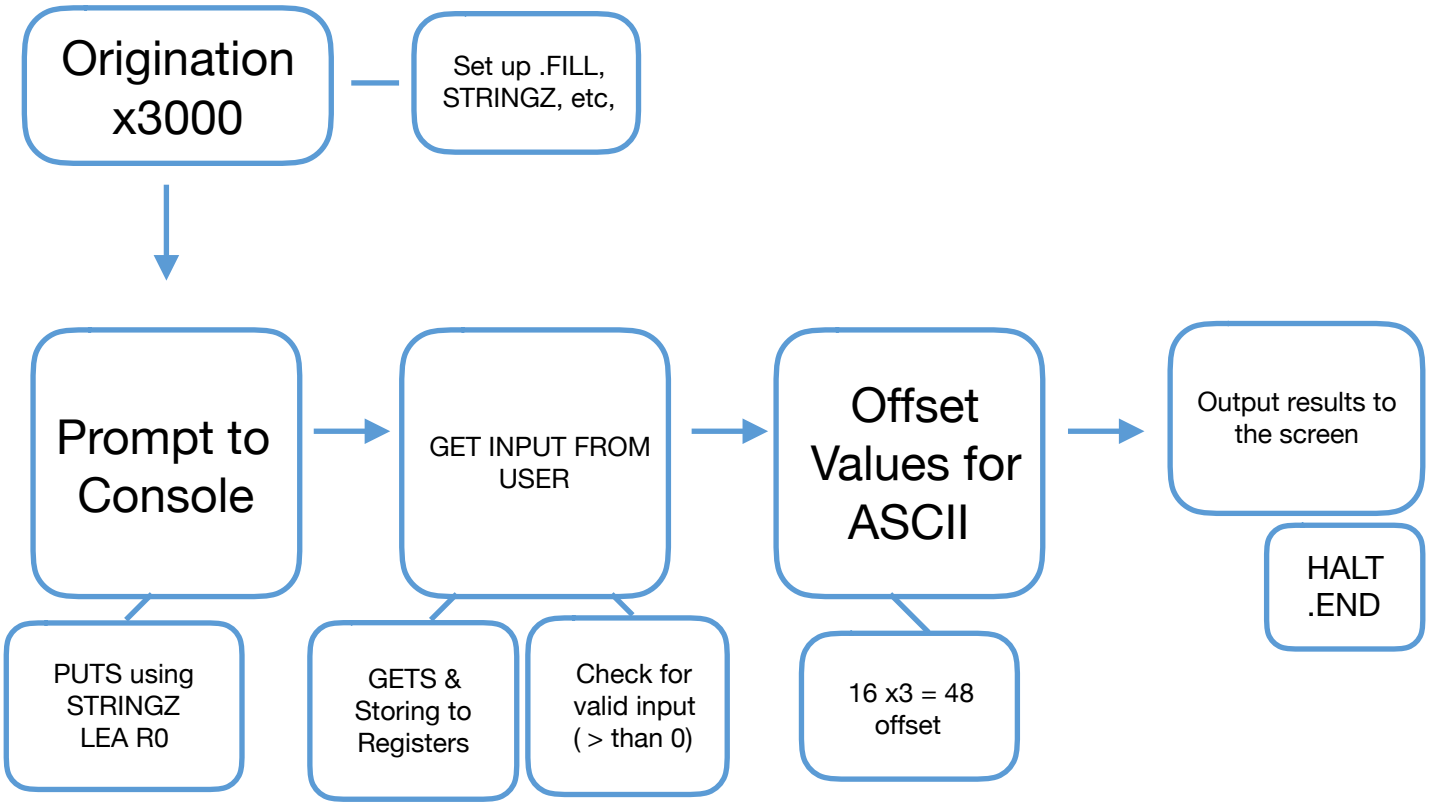
4.2.4 Error handling should be implemented and the application should be able to handle all run time errors.

4.3 Flow Chart and Pseudocode.

1. Start with the origination label and simple registers for the base of the project.
2. Create a prompt section for later
3. Include puts and getc as well as save registers
4. Use the JSR DIV subroutine
5. Stack/unstack and Push/Pop
6. Branches
7. Display PUTS
8. Prompt stringz " Enter the five scores" – 52, 87, 96, 79 and 61
9. Decode?
10. .fill registers
11. .END

*Pseudocode below

MAIN CALLING PROGRAM:



SUBROUTINES: