

CIS11 Course Project Part 1: Documenting the Project

Fill in the following areas (purple).

Introduction

1.1 Purpose

The purpose of this LC-3 program is to get user input for 5 test scores and output the average, highest and lowest values. The program will be made using stacks and arrays to manipulate and retrieve data during calculations. This helps the user understand and make decisions based on the scores in a simple way.

1.2 Intended Audience and Users

The primary audience/user for this program will be students and teachers hoping to analyze test scores to provide an informed outlook on their progress.

1.3 Product Scope

The intention is to provide an easy to use tool for calculating important statistics about certain test scores.

1.4 Reference

Source Documents for the Program Requirements and Specification

The specifications of the project include containing the proper labels and functions within their proper locations. In addition labeled subroutines will be able to carry out correct addition and multiplication within the designated registers. This program will use a stack and an array to properly handle user values and data manipulation. Users will input values ranging from 0-100, 5 times to initiate the average, max, and min calculations. They will be able to see their inputs to verify that their inputs are correct. The output will notify the user of these results on the terminal.

2. Overall Description

2.1 Product Perspective

Primary program objectives: get 5 test scores (inputs) from the users. Then, displays (outputs) the minimum, maximum, and average grade with letter grade (A = 100-90, B = 90-80, C = 80-70, D = 70-60, F = 60-50)

Goals: Calculates all the inputs from the user to find the minimum, maximum, and average. Lastly, we evaluate it with the letter grade, and then display it on the screen.

2.2 Product Functions

The overall description of functionality:

Functionalities:

User Input and Conversion:

- Prompt the user to enter test scores.
- Convert the entered values to decimal format.

Array Storage:

- Store the converted values in the appropriate locations within an array.

Stack Manipulation:

- Move the values from the array into a stack data structure.
- Comparison and Tracking:
- Pop values from the stack and compare them to determine the highest and lowest scores.

Output and Storage:

- Store the highest, lowest, and other relevant values in proper labels or variables.
- Display the calculated results or output to the user.

Technical functionality

What are the technical functions of the program? Subroutines and operations.

- Decode and Encode subroutines to convert from ascii to decimal and vise versa
- Mult and Division subroutines to get average and decimal scores from ascii values
- Grade subroutines to convert decimal values to a corresponding grade
- Stack operations such as pop and push
- Branching operations to compare which values are bigger or smaller
- Splitting functions to get the 1st and 2nd digit for comparisons

2.3 User Classes and Characteristics

Students learning Assembly programming who are implementing learned material through a hands-on approach. This is a collaborative project so ideas and knowledge are shared among the group. The main focus of this task is to gain practical experience in LC3 and implement what has been learned this far.

2.4 Operating Environment

The application will be operated on Windows 10 or any system compatible with LC-3 architecture. The development will be done on the Windows LC-3 Source Code Editor and tested on a LC-3 simulator. The application should be through a LC-3 Simulator by running the proper .asm files or code.

2.5 Design and Implementation Constraints

The Program may not exceed greater than 100 since this will require additional calls within the program. Users cannot input more than 5 test scores and the following calculation will not be in decimal form.

2.6 Assumptions and Dependencies

LC-3 Simulator with either .asm file or source code

3. External Interface Requirements

3.1 User Interfaces

The user can run it using the LC-3 Simulator. The program will then ask the users for inputs then displays the result.

3.2 Hardware Interfaces

Any computer types can run it, including Microsoft and Apple.

3.3 Software Interfaces

Do not require.

3.4 Communications Interface

No internet required. Only the program file and run it on the LC-3 Simulator.

4. Detailed Description of Functional requirements

4.1 Type of Requirement (summarize from Section 2.2)

Grade Function:

Purpose: This function is responsible for determining the grade corresponding to a decimal number.

Inputs: Decimal number as input.

Outputs: Grade (A-F) as output.

Encode and Decode:

Purpose: These functions convert between ASCII and decimal values.

Inputs: ASCII character or decimal value as input, depending on the specific function.

Outputs: Decimal value or ASCII character as output, depending on the specific function.

Stack:

Purpose: The stack is used to store values from an array using a loop.

Inputs: Array of values as input.

Outputs: Values stored in the stack.

Mult and Divide:

Purpose: These functions carry out basic arithmetic operations (multiplication and division) and store the result in a designated register.

Inputs: Two numbers for multiplication or division as input.

Outputs: Result of multiplication or division stored in the designated register.

Split:

Purpose: This function splits the first and second digits of a whole number and stores them in separate registers.

Inputs: Whole number as input.

Outputs: First and second digits of the number stored in separate registers.

Compare:

Purpose: This function compares the digits obtained from the Split function and stores the results of the highest and lowest numbers within registers.

Inputs: Digits obtained from the Split function.

Outputs: Highest and lowest numbers stored in separate registers.

4.2 Performance requirements

The expected performance would typically be efficient and fast, as the program is performing simple calculations on a small set of values from the user.

4.3 Flow Chart OR Pseudocode.



