Jacob Holland (Team H, Option B)
CIS-11 (22953)
Prof. Kasey Nguyen

CIS11 Course Project Part 1: Documenting the Project

1. Introduction

1.1 Purpose

Calculate & display the maximum, minimum and average scores of five test scores entered by the user.

1.2 Intended Audience and Users

Teachers, professors, educational personnel, students.

1.3 Product Scope

A tool for users that need to calculate & analyze test scores, supporting assessment performances.

1.4 Reference

LC3 Simulator documentation.

2. Overall Description

2.1 Product Perspective

The objective will be to include statistics from the user's test scores and provide letter grades based on the scores. An array of integers will be used for the five user scores and ASCII characters will be used to display the letter grades.

2.2 Product Functions

Prompt user to input test scores, store test scores in an array, calculate the minimum/maximum/average scores, display scores and their letter grades.

Technical functionality

Input/Output operations for user prompt and score display, arithmetic calculations for average (Addition & division), relational functions for finding minimum & maximum, conditional functions to find letter grade and perform loops, ASCII conversion to display letter grade, stack management, data movement.

2.3 User Classes and Characteristics

N/A

2.4 Operating Environment

LC3 Simulator (Version 3.01) running on Windows 10/11 OS.

2.5 Design and Implementation Constraints

May be limited by precision of integer arithmetic operations.

2.6 Assumptions and Dependencies

Assumes users are used to the process of inputting for scores. Dependent on a program that can run Assembly code for I/O and results.

3. External Interface Requirements

3.1 User Interfaces

Console.

3.2 Hardware Interfaces

Keyboard.

3.3 Software Interfaces

LC3 Simulator/program that can run Assembly program.

3.4 Communications Interface

N/A

4. Detailed Description of Functional requirements

4.1 Type of Requirement

Output (User prompt for scores, display numbers/score/letter grade), input (Obtain integers from user), GRADE subroutine (Determine the letter grade of a score), INARRAY subroutine (Store integers into array), MINIMUM subroutine (Find lowest score in array), MAXIMUM subroutine (Find highest score in array), AVERAGE subroutine (Calculate average of scores), DIVISION subroutine (Divide sum of scores by 5).

4.2 Performance requirements

To provide close, accurate response in performing and displaying calculations.

4.3 Pseudocode

```
.ORIG x3000

1. ENTER TEST SCORES INTO ARRAY OF 5 ELEMENTS
INARRAY FOR-LOOP (5 times):
    Start counter at #5
    Output: "Enter test score: "
    Input: Two-digit number
    Number gets stored into array
    Subtract counter by #1
    If counter equals 0, branch out of loop
    Else, repeat loop

2. OUTPUT SCORES AND LETTER GRADES
2a. OUTARRAY FOR-LOOP (5 times):
    Start counter at #5
    Output: Test score (Counter - 1 element in array),  " - ",  Letter grade for test score
        2b. FIND & OUTPUT LETTER GRADE
        GRADE SUBROUTINE
            ASCII = x46  ('F')
            Subtract 59 from test score
            If test score is negative or zero, return ASCII (x46 = 'F') and branch out of
            GRADE
            Else, ASCII = ASCII - x2 (x44 = 'D')
            2c. DCBA FOR-LOOP (3 times):
                Start counter at #3
                Subtract 10 from test score
                If test score is negative or zero, return ASCII, break out of loop
                Else, subtract counter by 1
                Subtract ASCII by x1
                If counter reaches 0, return ASCII (x41 = 'A'), break out of DCBA and GRADE
                Else, repeat DCBA loop
    Subtract counter by 1
    Repeat OUTARRAY loop
```

3. FIND & OUTPUT MINIMUM SCORE

3a. MINIMUM SUBROUTINE

```
    Store first element of test score array into MIN
    3b. MINNUM FOR-LOOP (4 times)
        Set counter to -4
        If test score[Counter+5] is less than MIN, store test score[Counter+1] into MIN
        Add 1 to counter
        If counter is negative, repeat loop
        If counter reaches 0, end loop
    Output: "Minimum score: ",  MIN,  " - ",  Letter grade for MIN (GRADE SUBROUTINE)
```

4. FIND & OUTPUT MINIMUM SCORE

4a. MAXIMUM SUBROUTINE

```
    Store first element of test score array into MAX
    4b. MAXNUM FOR-LOOP (4 times)
        Set counter to -4
        If test score[Counter+5] is greater than MAX, store test score[Counter+1] into MAX
        Add 1 to counter
        If counter is negative, repeat loop
        If counter reaches 0, end loop
    Output: "Maximum score: ",  MAX,  " - ",  Letter grade for MAX (GRADE SUBROUTINE)
```

5. CALCULATE & OUTPUT AVERAGE SCORE

5a. AVERAGE SUBROUTINE

```
    Add first element of test score array into AVG
    5b. SUMNUM FOR-LOOP (4 times)
        Set counter to -4
        Add next element of test score array into AVG
        If counter is negative, repeat loop
        If counter reaches 0, end loop
    5c. DIVISION SUBROUTINE
        ...
    Store result of DIVISION into AVG
    Output: "Average score: ",  AVG,  " - ",  Letter grade for AVG (GRADE SUBROUTINE)
```

HALT

VARIABLES

COUNTER
GRADE (ASCII character)
SCORE
MIN
MAX
AVG

.END