

## CIS11 Course Project Part 1: Documenting the Project

Group: We're In This Together

Names: Alliyah Munir, Ariana Nelson, Lissette Serrato, Isaac Martinez

### Project Choice: Option 1: Option A: Bubble Sort

Ideas for Project

- use one array/pointer
- Swap the value one by one
- To do comparison: use 2s' complement or subtraction (if you subtract and its still large, then its bigger and goes to front)
- If its zero, goes to BRz zero and if its negative, the program exits due to improper input
- When turning in: Send screenshots with diff values and show it works
- User interaction is necessary. Ask user to input values and then they input it( make sure program is ascii converted).
- Due in two weeks(may 19)
- if improper input is processed, maybe the prompt will be displayed again(using loops, include above)

## Introduction

### 1.1 Purpose

This document details the functionality and requirements.

- Requirements: Create an LC-3 program that allows users to input 8 numbers from 0 – 100, with an output of those 8 numbers in ascending order
- Functionality: Using appropriate address, labels, comments, subroutine, branching, stacks, branching, and ASCII conversion operations, we can allow the user to input 8 numbers from 0 – 100 and present an output of those 8 numbers in ascending order
- Purpose: To allow users to sort 8 numbers, ranging from 0 – 100 in ascending order

### 1.2 Intended Audience and Users

- Intended Audience: Programmers who would like to sort 8 numbers at a time in ascending order, ranging from 0 – 100
- Users: Developers/Programmers, Evaluator/Tester

### 1.3 Product Scope(Intentions)

- Assist programmers who need an efficient way to sort numbers ranging from 0 – 100 in ascending order. This program will allow users to input 8 numbers at a time and provide a quick output, saving them time.

### 1.4 Reference

#### Source Documents for the Program Requirements and Specification(References)

- LC-3 Simulator (to run the program)
- Reference on Github: <https://wchargin.github.io/lc3web/>

- LC-3 Text Editor, in order to insert input and view output

#### Companion Application Requirements Documents (If applicable)

- LC-3 Editor (location to insert input and view output)
- LC-3 Simulator (to run the program)
- Keyboard (directly input values)
- Mouse/Touchpad (to locate the program and test it)

#### (Applications to review program)

What other documents should be reviewed with this document?

- Program Requirements File, titled "CIS11 Course Project Part 1: LC-3 Project"
- Flowchart
- Pseudocode

## 2. Overall Description

### 2.1 Product Perspective

Primary program objectives(break down your intro..what is the goal, what is the process of search(linear/binary), how is it sorting)

-In order to achieve our goal of allowing users to input 8 numbers ranging from 0 -100 and place them in ascending order, we will use bubble sort algorithm

- To begin sorting, we need the program to reorganize the numbers( located in array) to be in ascending order, which can be done through a search process

-search process includes a binary or linear search, with binary search being more advantageous due to its speed and ability to begin comparing elements starting from the middle of the array

-Using an array and branching into subroutines is also necessary when programming, in order to ensure that the inputted numbers are placed in ascending order

-Branching can be done by testing each of the 8 numbers inputted, on whether they are positive, negative, or zero

-If positive, the program will continue processing, in order to verify that the numbers are in ascending order. If negative, the program will HALT and show ERROR. If zero, the program will automatically place the zero in front of all the other inputted numbers.

## 2.2 Product Functions

The overall description of functionality:

Subfunctionality and how it brings back the main purpose of program (talk about how things are returned)

Main/Technical Function: Prompt the user to input 8 numbers ranging from 0 –100, receive input of 8 numbers, and provide an output of those numbers in ascending order

SubFunctions:

- Display a prompt asking user to input 8 numbers, 1 at a time, ranging from 0 - 100
- Next, we will place the inputted integers in an array and properly organize the data
- Using sub-routines, we can separate numbers into branches depending on whether they are equal to zero, positive, or negative
- If input is not within the guidelines of the prompt, error message will display, and program will halt
- Once the inputted numbers are arranged, an output of the 8 numbers will be presented in ascending order

## 2.3 User Classes and Characteristics (Who are involved in this development process? Include business and technical personnel and their tasks.)

-Those involved in development process: Tester, Evaluator, and Developer

-Task of Tester: Analyze the program to test if program produces appropriate output, how the program reacts to incorrect input, and if the speed of the program is reasonable. Use the program for any purpose involving the arrangement of 8 numbers at a time, ranging from 0 – 100 in ascending order.

-Task of Evaluator: Analyze the program to test if it produces appropriate output, responds with a proper output with incorrect input, and if code in LC-3 editor is appropriate in terms of branches, fills, array style, search algorithm, conditional operations, pointers, ASCII conversion operations, system call directives, with appropriate labels and comments.

-Task of Developer: Ability to form a program with prompt asking user to input 8 numbers ranging from 0 – 100 and provide an output of those 8 numbers in ascending order. At the same time, developers need to fulfill program requirements, form a flowchart and pseudocode for this Bubble Sort Program.

**Operating Environment** (What type of system will the application be operated on? Operating system? System types? Development platform? )

-Operating System (Development Platform): LC-3 Editor(to view code), LC-3 Simulator(to run the program and produce an output)

-Operating System (System Types): Any including MAC OS, Android, Windows 10, Chrome OS

-Operating System (Browser Specific such as Google Chrome, Firefox, Internet Explorer): N/A

## 2.4 Design and Implementation Constraints(Limitation to application)

-Can't input numbers above 100 (or 100 years of age). Furthermore, this survey needs to have 8 numbers at one time, and speed will be constant due to the same number of inputs the program can take at once (8).

-Input Limitation: Users are only able to input 8 numbers at once, which must range from 0 – 100. If it is not in that range, program will halt, exit and present an error message.

-Performance Level: The program will only prompt the user to enter 8 numbers ranging from 0 -100 and will retain the input, to produce an output of those 8 numbers in ascending order

-Speed/Efficiency: The program will have a constant speed and its ability to present a prompt, obtain an input, and present an output will be persistent

## 2.5 Assumptions and Dependencies(expectations before-hand)

-User has the ability to read/understand the English Language

-User can save code located in the LC-3 Editor and run the program in LC-3 Simulator.

-User has the ability to enter 8 number ranging from 0 – 100 in LC-3 Simulator

## 3. External Interface Requirements

3.1 **User Interfaces**(what/how to input data and what/how will there be an output)

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

-User will be presented with a prompt stating, "Input 8 whole numbers, one at a time, ranging from 0 – 100"

-User will then input the 8 numbers within the appropriate range

-If user does not input the 8 numbers appropriately (numbers being too large or small), program will exit/halt the processor

-If user inputs the 8 numbers appropriately, the program will provide an output of those 8 numbers in ascending order

3.2 **Hardware Interfaces**

-Laptop, Tablet, or Desktop Computer with a monitor, keyboard, and mouse

3.3 **Software Interfaces**(What type of software will the application require to run?)

-LC-3 Simulator(run the program to provide input and achieve an output of 8 numbers, ranging from 0 – 100, reorganized in ascending order)

-Lc-3 Editor (To view the LC-3 code and save the program)

3.4 Communications Interface (Does your application require web, Internet or network connectivity? What type of network connection?)

-Browser(Not Required): N/A

Internet Connection: Options include Cellular, Wireless, and/or Satellite

-Internet/ Network connection: Options include WIFI (Wireless Fidelity), Wireless Connection, Virtual Private Network (VPN), Local Area Network (LAN), and/or Direct Network

## 4. Detailed Description of Functional requirements

**4.1 Type of Requirement (summarize from Section 2.2)( What are the functions? purposes? Inputs? Outputs? Data? Where is the data stored (internal or external to the application)?)**

**Functions:** Prompt the user to input 8 numbers ranging from 0 –100, receive input of 8 numbers, and provide an output of those numbers in ascending order. This should be done using an array, branches, stack, system call directive, and appropriate ASCII conversion operations.

**Purpose:** To give programmers the ability to input 8 numbers at once ranging from 0 – 100, and be presented with an output of those 8 numbers in ascending order

**Input:** User is prompted to enter 8 whole numbers that range from 0 – 100, one at a time. The user will then input 8 whole numbers of their choice using a keyboard and mouse

**Processing:** The program will then process the 8 whole numbers inputted and place it in an array. Then, using binary search, it will place those 8 numbers in ascending order. If the program receives a number greater than or less than 0 – 100, the processor will halt and exit the program

**Output:** The 8 whole numbers inputted earlier, will be reorganized into ascending order.

**Data:** Obtains an input of 8 whole numbers ranging from 0 - 100, and provides an output of those 8 numbers, from least to greatest

**Data Storing:** Will be stored externally, allowing the user to view the output and provide input by running the LC-3 Simulator containing the LC-3 Editor's file.

#### 4.2 Performance requirements

**What is the expected performance level of the program?(Edit: how does it handle error?)**

**Expected Performance Level:** The program will begin by presenting the user with prompt stating, "Input 8 whole numbers, one at a time, ranging from 0 – 100." The program will then take in the user's input and provide an output of those 8 numbers in ascending order. **If the user inputs numbers incorrectly, the program will halt and exit the program.** Overall, the database should be scalable with a capacity to hold any 8 numbers ranging 0 – 100. Also, the program should continue to produce correct output when run multiple times.

**Accessibility:** The program should be accessible to those who have an LC-3 Editor, an LC -3 Simulator, reliable network connection and a desktop.

**Speed:** The program's speed should be constant if the user provides appropriate input. However, if the user does not provide correct input, the program may lag. One must also have reliable internet connection while running this program.

**Error Handling:** **Run time errors are handled by halting the program, then exiting.**

#### 4.3 Flow Chart and Pseudocode(break each subtask into smaller tasks.



