# CIS11 Course Project Part 1: Documenting the Project

**Introduction**

* 1. **Purpose**

The purpose of this project is to create a program that implements a bubble sort for a list of numbers. This project is also intended to help students learn how to code in assembly and understand the basics of LC-3.

* 1. **Intended Audience and Users**

The primary audience/user is anyone who wants to sort a set of numbers or learn how to code in LC-3.

* 1. **Product Scope**

This program’s intention is to sort values inputted by a user using a bubble sort.

* 1. **Reference**

**Source Documents for the Program Requirements and Specification**

Reference Project requirements and LC-3 specifications.

1) implementations of Bubble Sort in C++ to be converted to assembly

<https://www.geeksforgeeks.org/bubble-sort/>

2) Code used in assignments/labs/Class lectures

3) ASCII Table

<https://www.asciitable.com/>

4) TRAP Vectors

<https://jupyter.brynmawr.edu/services/public/dblank/CS240%20Computer%20Organization/2015-Fall/Notes/LC-3%20I-O.ipynb#2.3-OUT---Trap-x21>

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

What other documents should be reviewed with this document?

1. Project.asm file
2. Documentation file
3. Github repository

**2. Overall Description**

**2.1 Product Perspective**

Primary program objectives

This program looks to use assembly logic in order to sort numbers using a bubble sort. While in C++ this is done with very little code assembly takes many more lines due to the way were are processing the data.

* 1. **Product Functions**

**The overall description of functionality:**

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

The program starts by asking the user to input a list of 8 numbers to be sorted. From there a function to get the numbers will be implemented. This function will contain a loop in order to all 8 numbers to be stored. These stored values will then be passed into the bubble sort function which will sort the data. A function that prints the list of sorted values will then be called in order to display the list on the console for the user to read.

* 1. **User Classes and Characteristics**

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

* 1. **Operating Environment**

This program will run using LC-3 assembly language

* 1. **Design and Implementation Constraints**

Access to an LC-3 simulator is required however user can also use online emulators. If using an online emulator an internet connection is required to access the emulator. Other constraints is that because there are a limited number of register in LC-3 we are not able to make the program very complicated. This means that we are required to reuse registers and make sure that data transfer is efficient and effective. If these requirements are not met the program will run into errors when trying to process the data being passed into the program.

* 1. **Assumptions and Dependencies**

It is expected that the users of this program will have some previous knowledge of IDE use and programming knowledge. This will help the user understand how to use the program and how it works. We also assume that the user has access to an LC-3 simulator, however there are online emulators.

***3*. External Interface Requirements**

* 1. **User Interfaces**

The user will be in charge of choosing values to sort

* 1. **Hardware Interfaces**

Specify hardware interface – computer types? Terminal types?

* 1. **Software Interfaces**

Requires LC-3 simulator and text editor.

* 1. Communications Interface

If LC-3 is not installed on the user’s computer they will have to run the code on an online emulator. This will require connection to the internet.

**4. Detailed Description of Functional requirements**

**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)?**

The following functions are the main functions that I plan top use in my program:

(more functions will be required to implement the program however these are the main functions called)

**ASK4INPUT:**

**Asks the user for their inputs this section of the code will require a loop in order to ask for more than one number.**

**GETINPUT:**

**Allows the user to input values and stores them for later use. This function will have to be looped as well in order to get more than one number**

**BUBBLESORT:**

**Implementation of bubble sort on the list of inputted values.**



**used as reference for bubble sort**

**PRNTNUMS:**

**Function that prints the sorted list of numbers. This function will loop as well due to the way the numbers are being stored. I plan on using 3 registers to hold the characters of the inputted number. 100 will use 3 registers to store 1, 0, and 0.**

**4.2 Performance requirements  
 What is the expected performance level of the program?**

**This program is expected to accept 8 values to be sorted. The program should be able to sort any number between 0-100. This program is far from perfect and may still contain errors however it does do its job of sorting a list of numbers.**

**4.3 Flow Chart and Pseudocode.**

**Pseudocode:**

**.ORIG x3000 ;ORIGINATION ADDRESS**

**JSR ASK4INPUT ;jumps to subroutine**

**ASK4INPUT**

**A4ILOOP ;loop in order to ask for more than 1 num (loops 8 times)**

**Branch EXITLOOP**

**JSR GETINPUT**

**GILOOP**

**GETC**

**OUT**

**ASCII Offset**

**Checks for ascii value 48-57**

**Uses conditional branches in order to determine ascii value**

**Stores inputted values**

**JSR BUBBLESORT ;jumps to subroutine**

**BUBSORTLOOP**

**LOOPCOUNT**

**BSLOOP**

**INTARRYAY[COUNTER1] - INTARRYAY[COUNTER2]**

**Will iterate through conditions until all conditions for bubble sort are met**

**JSR PRNTNUMS ;jumps to subroutine**

**PRNTLOOP**

**Will loop 8 times in order to print numbers for sorted list**

**.Fill for all labels, variables, and strings**

**.End**

**Flow Chart:**

**Basic flowchart of program implementation:**

|  |
| --- |
|  |
|  |  |

**JSR ASK4INPUT**

**GILOOP**

**GETC**

**Store inputted values**

|  |
| --- |
|  |
|  |  |

**Exit function**

**Back to main program**

**b**

**JSR BUBBLESORT**

**LOOPCOUNT**

**BSLOOP**

**JSR PRNTNUMS**

**.END**

**Program ends**

**PRNTLOOP**

**.Fill for all labels, variables, and strings**