BUBBLE SORT PROGRAM

PHASE I: PROJECT PLANNING

- Team: A+sian Assembly
- Identified goals of the program
  - Given an unsorted array of numbers, the program will sort them
  - Various subroutine implementations to handle various aspects/subtasks of the program
  - Assigned team roles/contributions
    - All team members contribute to code
    - Joseph: Logic Implementation
    - Chloe: Documentation + Subroutines
    - Darren: Documentation + Subroutines

PHASE II: CODING AND DEBUGGING

- Using the pseudocode as the framework for our project, we split up the coding into various subtasks and handled them individually.
- Input - Intended to gather input and place into memory: wasn't able to get it to work
- Sort - Uses the classic bubble sort algorithm in order to sort code in memory
- Swap - A utility function for sort that swaps the position of numbers in memory.
- Output- Intended to go through the array and display each value: wasn't able to get it to work
- Multiply-by-10 - Utility function for input intended for input function, intended to multiply a number by 10
- Stack - each function uses simple pop and push operations to manage a stack of addresses to return to.
- During prototyping, multiple files were used to test out subroutines before fully implementing

PHASE III: REFLECTIONS

- LC3 is very low level and limited so it would have been more helpful to use more descriptive documentation and comments
- A high-level design that could be made with considerations of these limitations would be helpful.
- There was trouble with input/output due to limited understanding and experience on techniques of how to do it.