Louis Romeo
CSC 252
ASM 6 Report
12/6/2023

In ASM 6, I chose to implement the concept of a sorter, which takes user input from the keyboard in MIPS, then uses bubbleSort, and prints out the array in ascending order. To use the program, simply call the function sortInput, or run the test file I called asm6test.s. Once run the program will prompt you to answer in a single integer how long you want the array to be in the I/O tab. The program will prompt you to answer again if your answer was not an appropriate length. The program will next ask you about the numbers in the array, which must be inserted as a single integer also. Press the enter key after each integer to enter the next one until you've entered enough integers to the specified length. Once all the data is entered, the program will print out the original array with the values in the order they were entered, and then below it it will print the sorted array. A message will then ask you if you want to run the program again and repeat the above process. You can type Y or y for yes and N or n for no.

The first function, sortInput, does most of the leg-work and calls other helper functions when necessary. The getLen function is responsible for asking the user how long the array should be and saving the user input. Print_array only prints the array that is formed with the user input, and takes no parameters. The bubble_sort function handles all of the sorting of the values, and it does so by copying values from one index to another and sorts them in between. Finally, run-again utilizes MIPS syscall 54 to ask if the user wants to run the program again, which will call sortInput and start the process over again. I used three extra syscalls in order to make this program operate functionally. I used syscall 5 to get user integer input from the keyboard, syscall 54 which is a dialogue input string (to ask the user to run_again), and syscall 17 which is exit. The most commonly used syscall is 5 because it is called every time user input is needed.