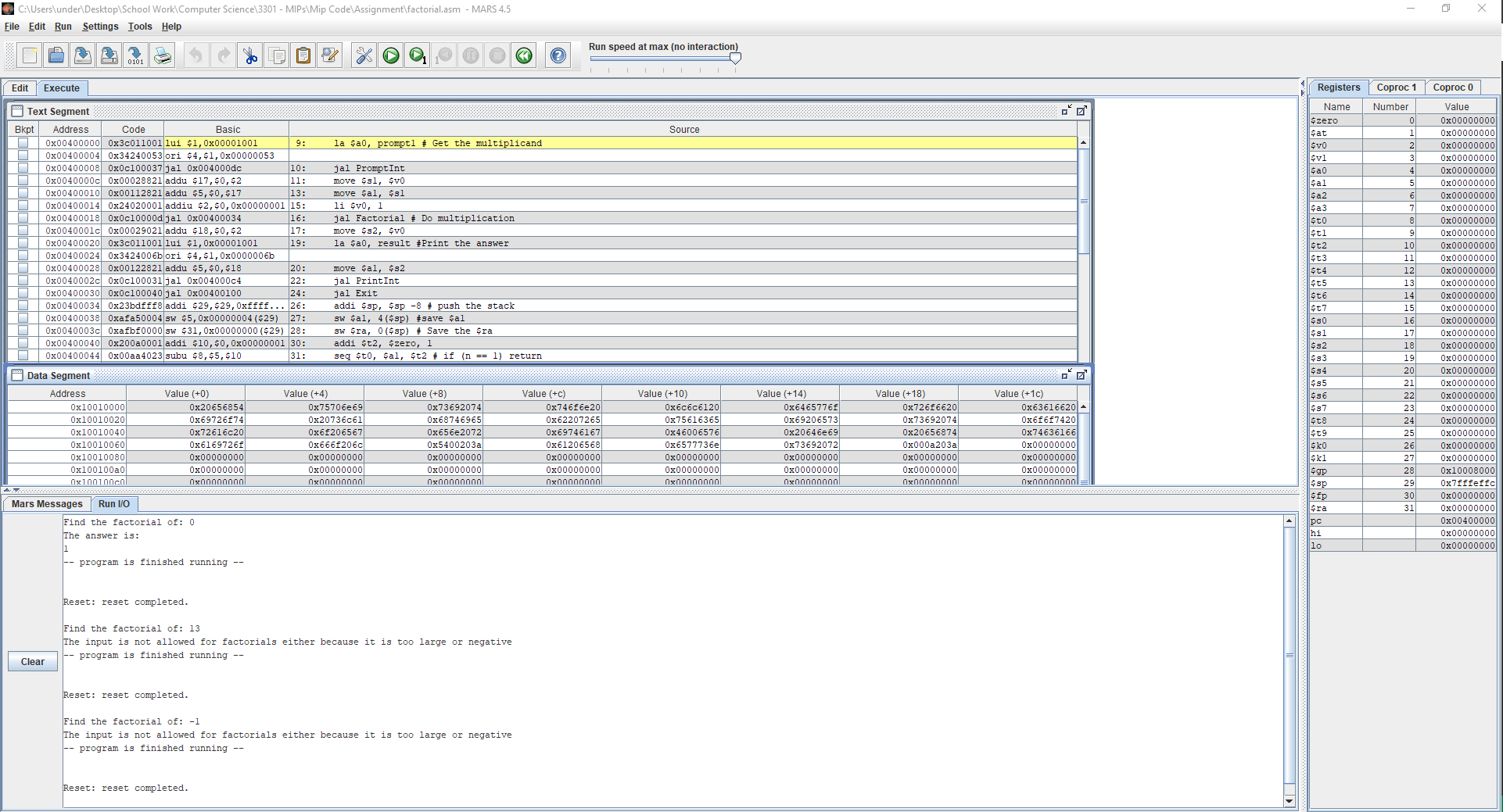
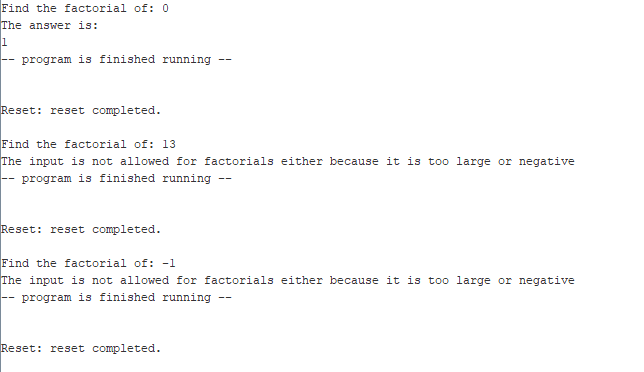
**Part 1: Factorial Program**

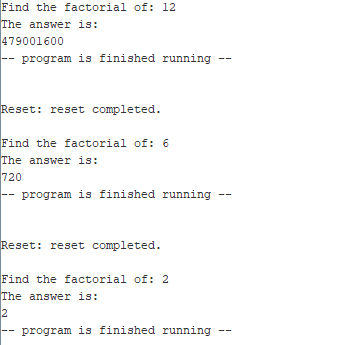


Above is a screenshot of my environment for the sorting program. I am using the MARS ide to produce this runtime.

I used sub procedure code obtained from *Introduction to MIPS Assembly Language Programming* in order to establish a working program for this problem. The edge cases for this problem were relatively straight forward. The only requirement to handle them were a series of branches. Specially ones detecting an input less than 0 and greater than 12. In the case that the initial value was 0 the program immediately returns the base case for the factorial, 1.

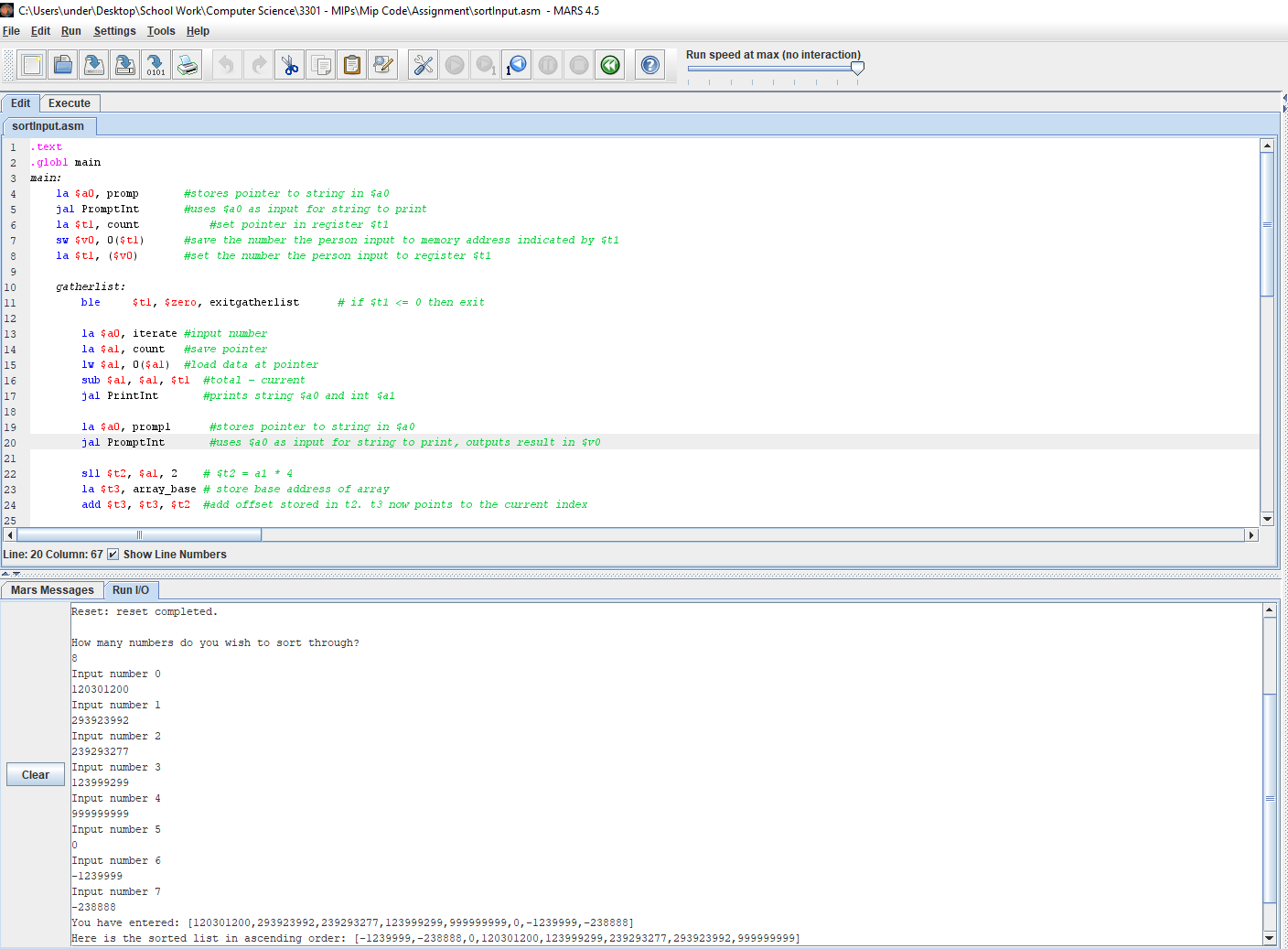


Here is a closer look at the terminal displaced on the previous page. It demonstrates the edge cases discussed earlier.



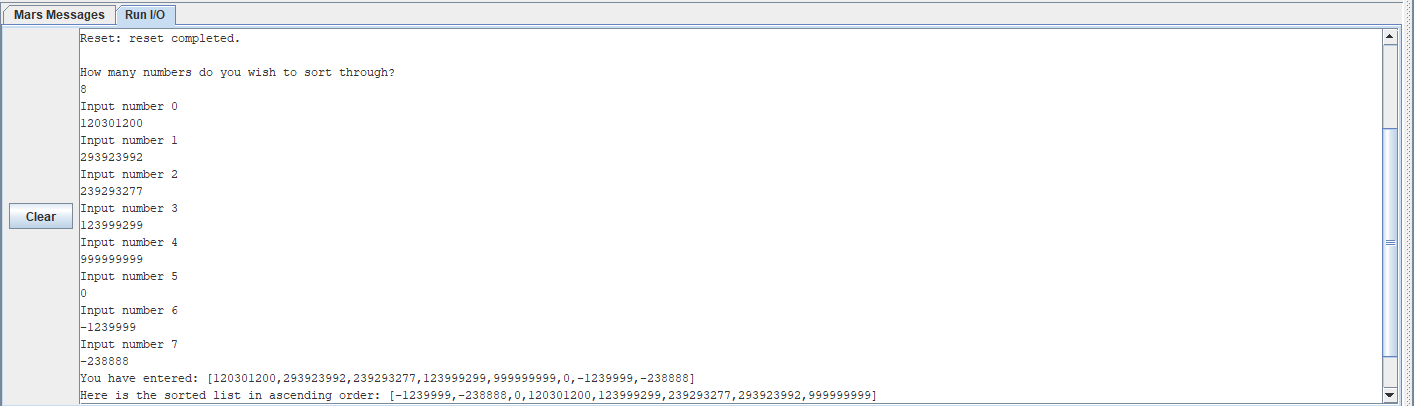
Finally, here are some examples of the program executing on regular values of n.

**Part 2: Sorting Program**

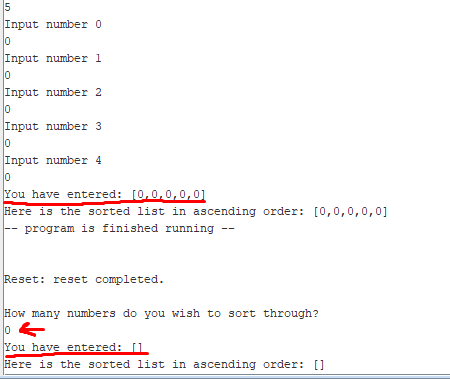


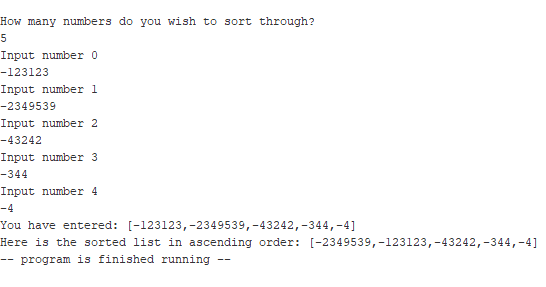
Above is a screenshot of my environment for the sorting program. I am using the MARS ide to produce this runtime.

I used sub procedure code obtained from *Introduction to MIPS Assembly Language Programming* in order to establish a working program for this problem. The difficulties I encountered were with allocating space for the array base address data to some arbitrary value, n. Therefore, I manually limited the data to 100 entries. I did not do extensive edge case testing for invalid entries. For example, if a letter is entered; the expected datatype is an integer so the program will throw an exception. This is not indicated specifically under the list of edge cases. So specifically, I will demonstrate the listed edge cases.

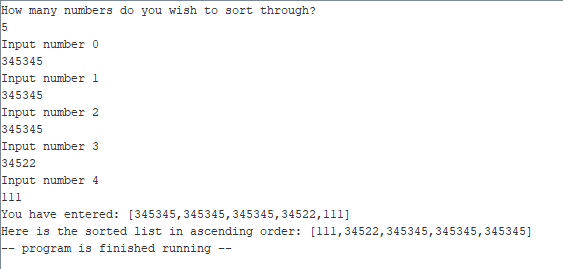


Here is a closer look at the terminal, I chose to sort 8 numbers, the maximum is arbitrarily 100. This execution demonstrates large negative and positive values correctly ordered with 0 included in the dataset. If an integer is larger than the maximum held in 32 bits then the program will throw an exception.

  
This is an example of an empty array, both in size and sum.



This is the output with only negative integers.



And finally, only positive output.