EE 2310 Experiment #5: A More Complex Assembly Language Loop Program

1. **Introduction:** Experiment 5 furthers builds on the skills developed in the EE 2310 assembly language homework. Although Labs 1-4 were done in the EE 2310 lab room, teams will work outside the lab on Experiments 5 and 6. Student teams should work together.

2. **Goal of this exercise:** Gain experience in designing a more sophisticated loop program. The focus is to make a sorting algorithm into a simple loop. A series of random letters, input from the keyboard, will be alphabetized and output it to the console in alphabetized order. The program provides further practice using syscalls 4 and 8 and the ".space" directive.

3. **Experimental Equipment List:** The following items are required:
   - **Pervin text for reference.**
   - **SPIM installed on your PC.**

4. **Pre-Work:** Do the following:
   - **Make sure that you are up-to-date in software reading assignments.**
   - **Read the program description below. Make sure that you clearly understand it.**
   - **Flow-chart the program, if necessary, to make sure that you understand its execution.**
   - **Lab partners should work on the program together as much as possible.**

5. **Program Description:** Write a program that does the following:
   - **Alphabetizes a list of small (non-capital) letters. For consistency, make the string 12-15 random small letters. Label the sequence "string 1" in the data statements, followed by a ".space" directive to reserve 20-30 byte spaces (extra room in case you input a few extra characters, and so that you will still have a null-terminated string, since ".space" clears all byte spaces to 0).**
   - **Inputs the string from the keyboard using syscall 8 and places it in the space labeled "string 1."**
   - **Alphabetizes the string using a loop and outputs the alphabetized string to the simulated console.**
   - **Removes any characters that are not small letters (capitals, punctuation, numbers, etc.).**
   - **Outputs the ordered letters, preceded by a statement: "The alphabetized string is: "**

6. **Comments About Program and Hint:** This program can be done with a recursive loop, as demonstrated in class. However, there is an easier approach. Start by looking for "a's." If there are any "a's," store them in a new string (labeled "string 2"). Then compare to "b," "c," etc. Such a loop will also let you assure that each character is a-z. If not, throw it away and do the next compare. When you get to a null, stop, since your ".space" command has assured that the string is null-terminated. Although you have to go through the input letters 26 times (a-z), the loop is so fast that this is not a problem. You can do a recursive sort/compare such as in the homework problem noted above (only going through the string of letters once, but going back-and-forth a great deal!), but the approach described above is easier to implement.
7. **Experimental Procedure:** Note: For Experiments 5-6, send the completed program to your TA as an email attachment. The overall procedure:
   - **Review the program specification and diagram the program flow.**
   - **Compose the program. You can work separately at first, but be sure to work together as well. When the program runs, make sure that the output string is correctly alphabetized.**
   - **When you are finished, submit a single copy in NotePad to your TA via email.**

8. **Laboratory Report:** A laboratory report is not required for lab software programs.