Project Description
Due: See notes below

Description
You are to implement a string manipulation package using MIPS and test it on SPIM simulator. The design uses 32 memory locations that point to strings or NULL as well as 32 integer locations. Use the symbols ls0 – ls31 and in0 – in31 to represent the string pointers and integer variables, respectively. These locations will represent the operands for a limited instruction set that you will design for this project. The instruction set should include all the functions listed below as well as any instruction that you deem appropriate to complete the instruction set such as transfer instructions, duplication instructions, etc…Use 32-bit instructions to represent the operation and the operands. Follow MIPS instruction set design guidelines.

String Representation
A string is represented using a null-terminated one-dimensional array. For the purpose of the assignment, a string has 1024 characters maximum.

List of Functions
For the package you have to implement the following functionality.

**STRLLEN**
This instruction should take a single argument that is a pointer to null-terminated string, calculate the length of the string, and returns the length of the string not including the terminating ‘\0’ in another operand.

**COUNTCHAR**
This procedure counts the occurrence of a particular character in a null-terminated string.

**COUNTWORD**
This procedure counts the number of words in a null-terminated string.

**STRCPY**
This procedure copies string into another. It takes a pointer to null-terminated string1 and another pointer. It copies string1 to the memory pointed to by string2.

**REVERSE**
This procedure reverses a string.

**STRCAT**
It concatenates string2 onto the end of the string1. It takes a pointer to null terminating string string1 and another pointer to null terminating string string2.
**STRCMP**
It compares the contents of string1 with that of string2. It reruns a value of type int; 0 if identical; positive if string2 would occur before string1 in the ordering given by ASCII character set, otherwise negative.

**STRSTR**
Find the first occurrence of string2 as a sub string of string1. It takes a pointer to null terminating string string1 and another pointer to null terminating string string2. It returns a pointer to the first character of the first occurrence of string2 in string1 or NULL if string2 doesn’t occur in string1.

**EDITSTR**
This procedure replaces a character of a string by another string.

**Notes**

?? Due dates:
- Two teammates maximum per project. Please email the GTA the members of your selected group by **Monday 12th of March 2001**.
- Phase I: Submit the complete design of the instruction set, operands, memory locations reserved for the project, etc… by **Friday 30th of March 2001**.
- Phase II: Submit the complete project including implementation by **Friday 27th of April 2001**.

?? After the submission, a demonstration period will be assigned per group to run and test their program.

?? You are required to submit a well-commented assembly code, and fully-document your design and implementation in a project report.