Assignment 6

You are going to apply your knowledge of processing character arrays in MIPS to complete this assignment. For this assignment, you will write procedures to convert strings and you will call these procedures based on input from the user.

- **toup** - to convert a null terminated ascii string to all uppercase letters. Do not convert characters that are already uppercase. Count the number characters that you convert to uppercase. Pass the memory location of the from string to convert in $a0. Pass the memory address of the string **to copy the converted string** into in $a1. Return the number of characters that were translated in $v0.
- **tolow** - convert a null terminated ascii string to all lowercase letters. Do not convert characters that are already lowercase. Count the number characters that you convert to lowercase. Pass the memory location of the string to convert in $a0. Pass the memory address of the string **to copy the converted string** into in $a1. Return the number of characters that were translated in $v0.

In your main,

- Ask the user for a string to convert to lowercase. Call the procedure to convert the string. Print the converted string and the number to characters translated.
- Ask the user for a string to convert to uppercase. Call the procedure to convert the string. Print the converted string and the number to characters translated.

Example output:

```
Enter a string to convert to lowercase: Hello World!
The converted string is: hello world!
The number of translations is: 2
Enter a string to convert to uppercase: Hello World!
The converted string is: HELLO WORLD!
The number of translations is: 8
```

Hints:

- Refer to Lecture 7 to refresh your memory on how to use bitmask to convert between lower and uppercase.
- You will also need to avoid translating any special characters (those not between A-Z or a-z).
- If you cheat on this assignment, you will receive a zero on this assignment and fail this course.

Challenge/Bonus Question (10 points):

- Write a procedure called **strcat**, which accepts the memory address of string 1, string 2, and string 3. The procedure concatenates string1 with string 2 and places the resulting value in string 3. In addition, the procedure will calculate the length of string 3 and return that value.
  `int strcat(char* string1, char* string2, char* string3);`
- From your main line, call the procedure **strcat** with the two strings already converted by tolower and toupper. After returning from the procedure, print string 3 and the length of string 3.

The following are required for all assignments and are included in the rubric for grading:

- You need to name your file as "LastName-Name-Assign7.asm" (Example: Talley-Michelle-Assign6.asm)
- Your program will need to have the exact output unless otherwise stated. ***Make sure to use spaces and newlines as required.***
- Your source needs to have comments that explain your implementation.
- Your procedures need to have comments.
- You need to make sure you exit your program and avoid calling procedures unnecessarily.
- You need to include the following set of comments at the top of your source code for all assignments.
  #Your Name
  #Assignment # (Example: Assignment #6)
- You need to submit your source code on blackboard.
- Please submit your files in a zip file named LastName-FirstName-Assign6.zip) and make sure you include any files that are used as includes in the zip file (Example: utils.asm).