Lab 4: str_to_upper

Due date: By the end of Thursday, 9/26/2019.

10 points

Objectives

The goal of this lab is to implement a recursive function with MIPS instructions that works on an ASCIIZ string. You will practice with the following skills:

- Write recursive or nested functions.
- Save and restore registers on stack.
- Process ASCII strings.

Tasks

First, read and understand the skeleton code. The code reserves a buffer of 128 bytes for a string `str`.

Implement the `strtoupper` function, which takes the starting address of a string as the only argument, converts all English characters in the string to uppercase, and returns the address of the input string. Your implementation MUST be recursive. A C implementation is listed on the next page.

The main function repeats the following tasks until the user enters an empty line.

1. Use a system call to read a line of user input and place it into buffer `str`. `str` is now a NUL terminated ASCII string. Assume the line is of 126 characters or shorter.
2. Call strtoupper to convert `str` to uppercase,
3. Print the converted `str`. All characters should be in uppercase.
4. Exit from the loop if the first character of `str` is `'\n'` (ASCII value 10) or a NUL.

Step 1 is provided in the skeleton code. **You need to complete Steps 2 to 4.**

Please pay attention to **the registers** you use in the function and follow the MIPS calling conventions.

Add brief **comments** to explain your code.

```
/* Comments
   The function converts all ASCII characters in string s to upper cases.
   s is the starting address of a string.
   The function returns the starting address of s,
   or you can think it as the address of the first character in s.
   char * means the address of a character.
*/
char * strtoupper(char s[])
{
```

```
	/* define local variables. */
	char c;

	c = s[0];

	/* empty string */
	if (c == 0)
		return s;

	/* convert the first character to upper case*/
	if (c >= 'a' && d <= 'z') {
		c -= 32;
		s[0] = c;
	}
	/* convert the remaining characters*/
	strtoupper(s + 1);
	return s;
}
```

```
Below is a sample session of running the entire program. Note that the
program terminates on an empty line.

CSE3666: Lab 4: YOURNAME (YOUR NetID)

abcde in cse3666.
ABCDE IN CSE3666.
Another line in the same session.
ANOTHER LINE IN THE SAME SESSION.
Press Enter only to exit.
PRESS ENTER ONLY TO EXIT.

-- program is finished running –
```

Deliverables

-Make sure to modify the line "CSE3666: Lab 4: **YOURNAME (YOUR NetID)"** in lab4.s with your name and netid.

-Submit revised lab4.s, which has your code and **comments**, in HuskyCT along with a screenshot of the outputs.

-To receive full credits, your code should use proper MIPS instructions/pseudo instructions for the tasks and follow the MIPS calling conventions.