Homework 8 C Programming - String Functions

Due Friday 11/1/2019 11pm

You have **2 parts** to complete this assignment. **Part 1**: you have to implement two functions **`mystrcmp()`** and **`mystrdup()`** that should do the same thing as the C library functions **`strcmp()`** and **`strdup()`.** **Part 2:** you have to create a document that is similar to the result of manpage to help people know how to use your function.

Download hw8handout.tar

Use instructions similar to homework 2 to download hw8handout.tar file, copy it to w204 machine's 311 folder, use `tar xvf` command to unpack it to hw8handout folder.

Functions to implement

You are to implement two functions as below. You are responsible for adding comments in your code to explain what the functions do. **You are not allowed to use any string library functions in your implementation** even though the starting code includes string library functions.

```
int mystrcmp(const char *s1, const char *s2) {
  return strcmp(s1, s2);
}

char *mystrdup(const char *s) {
  return strdup(s);
}
```

Edit/Compile/Test your C code

1. Compile your code

Assuming you successfully copied and unpacked hw3handout.tar file in ~/311 folder on your W204 account, the following command will help you compile the given C program. The original tar file contains a complete C program that compiles and works.

```
cse-p204inst11.cse.psu.edu 160% cd ~/311/hw8handout
```

You use **make** command in the hw8handout folder to compile all the .c files to create an executable called **mystring**.

Make sure that make does not generate any error/warning messages while compiling your C code. Any warning/error messages while compiling your code will result in a zero for this assignment. You must address these warnings/errors and remove all compiler warnings/error messages before you submit. To be safe, do the following before you submit to ensure no warnings/error messages were produced.
%make clean
%make

2. Test your code

We intentionally did not supply much code to test your implementation. You are responsible for finding ways to test that your implementation is conforming to what the real library functions should do. You can modify the code in main.c to add any testing you want.

3. Edit your code

Before you start, please make sure to write your name and email at the beginning of dict.c code to replace Prof. Wang's name and email.

```
// Author: Yanling Wang
// Email: yuw17@psu.edu
```

Edit your code in `mystring.c` with vim and save it and repeat step 1/2 to compile and test your code.

Always start with small changes before your compile and test your code.

Keep your code properly indented.

Add comment in your function to explain your algorithm for this assignment. Use block comments that are on its own lines. Do not use tail comments at the end of each line of the code.

Submit your C code (4 pts)

You are to submit only the **mystring.c** file to gradescope.

Gradescope is not responsible to provide a thorough test for your submitted code at the time of your submission, we will also read your code and grade your code with more tests as well as

looking out for warning/error messages from gcc. The testsuites we do provide is very basic and is meant to catch serious errors but it is up to you to find ways to thoroughly test your functions.

Submit your Professional Man-Page Document (6 pts)

You are to submit a pdf document **mystring.pdf** to gradescope.

Your document must include the following details for each of the function you implemented similar to what can be found in a man-page for a library function. Your document should also include the source code of your implementation.

- Name
- Synopsis
- Description
- Return Value
- (When applicable) Errors
- Source code
    - Must be properly indented
    - Must use **`Courier New Font, Size 12`**
    - Must include Block Comments for the source code.