

Task Details

Implement a shared library **libcs392_midterm.so**, which provides the APIs for **cs392_strcpy**, **cs392_strcmp**, and **cs392_strncat**. Detailed descriptions are as follows:

- **char *cs392_strcpy(char *dest, char *src);**
 - The **cs392_strcpy** function copies the string pointed to by **src**, including the terminating null byte ('\0'), to the buffer pointed to by **dest**.
 - This function returns **dest**
 - Please do not worry about the size of **dest** and **src**. They are big enough.
 - Please do not worry if **dest** and **src** are NULL. They won't.
 - Hint: to access the i-th character in **src** or **dest**, use: **src[i]** or **dest[i]**.
 - Hint: string stops at '\0';
- **int cs392_strcmp(char *s1, char *s2);**
 - The **cs392_strcmp** function compares the two strings **s1** and **s2**. It returns an integer **less than, equal to, or greater than zero** if **s1** is found, respectively, to be **less than, to match, or be greater than s2**.
 - Please do not worry about the size of **s1** and **s2**. They are big enough.
 - Please do not worry if **s1** and **s2** are NULL. They won't.
 - Hint: **Compare based on ASCII value of each character** until non-equivalence or end of either string. If two strings have the same prefix but they have different length, then the shorter one is smaller.
 - Hint: example of getting the ASCII value of the i-th element in **s1** or **s2**: **(unsigned char) s1[i]** or **(unsigned char) s2[i]**
- **char *cs392_strncat(char *dest, char *src, unsigned n);**
 - The **cs392_strncat()** function **appends** the **src** string to the **dest** string. This function **overwrites the terminating null byte ('\0') at the end of dest** with the first byte to append. Also **it adds a terminating null byte after the appending**.
 - This function will use at most **n** bytes from **src** (**excluding the NULL byte '\0'**); and **src** does not need to be null-terminated if it contains n or more bytes.
 - This function returns **dest**
 - Please do not worry about the size of **dest** and **src**. They are big enough.
 - Please do not worry if **dest** and **src** are NULL. They won't.

Requirement

1. Your project should include three source files **cs392_strcmp.c**, **cs392_strcpy.c** and **cs392_strncat.c**
 - a. cs392_strcpy.c defines function cs392_strcpy

- b. cs392_strcmp.c defines function cs392_strcmp
 - c. cs392_strncat.c defines function cs392_strncat
 - d. Please use the exact names as above. Otherwise you will not be able to use the provided testcase
2. Your project should include a header file "**cs392_midterm.h**" to provide prototypes (i.e., declarations) of the above three functions. This header will be used in the test-case program.
3. Use of standard c library functions is not permitted. For instance, strlen, strcpy, strcmp, strncat are not allowed to be called in your code.
4. Place "I pledge my honor that I have abided by the Stevens Honor System." as comment in the beginning of each "*.c" and *.h* file
5. Zip all the files into a cs392_midterm.zip and submit the .zip file only
6. Submit via Canvas
7. No late submission is permitted without approval

Test-case

A test program is provided at:

<https://drive.google.com/file/d/1xiwtFcwtd3wT8JVxRR3CZogXxEtCrz5u/view?usp=sharing>

A Makefile is provided at:

https://drive.google.com/file/d/15jeWbnb_1L4rX4PcBY1WeADwixAXd1w4/view?usp=sharing

How to do test:

- Download the program "cs392_midterm.c" and the "Makefile" from the above links
- Place "cs392_midterm.c", "Makefile", "cs392_strcmp.c", "cs392_strcpy.c", "cs392_strncat.c", and "cs392_midterm.h" in the same folder
- Run the exact following commands:
 - **make clean**
 - **make**
 - **make test**
- You will see how many case tests you pass for each function (eg., "You passed 4 test cases for cs392_strcpy")
 - cs392_strcpy at most 4 successes
 - cs392_strcmp at most 4 successes
 - cs392_strncat at most 2 successes
- Whenever you change your code, you must **re-run all the above three commands** if you want to re-do the test

Grading policy:

- No statement of “I pledge my honor that I have abided by the Stevens Honor System.” as comment in the beginning of your code: **Will not be graded**
- Total point: 110
 - 10 points are bonus, just in case the exams are tough.
 - If you get more than 100 points, you get 100 finally
 - If you get less than 100 points, you get the exact points
- Pass one test case (+10)
 - In total: $(4 + 4 + 2) * 10 = 100$
 - If your code re-uses c standard library functions (e.g., strcpy), you will not be able to get the points for the corresponding test-cases.
- Provide the right header file and avoid duplicated inclusion (+5)
- Comments for key code (+5)

Note:

If you want to test the Makefile without finishing your code:

Place

```
char *cs392_strcpy(char *dest, char *src){  
    return dest;  
}  
into cs392_strcpy.c
```

Place

```
int cs392_strcmp(char *s1, char *s2){  
    return 0;  
}  
into cs392_strcmp.c
```

Place

```
char *cs392_strncat(char *dest, char *src, unsigned n){  
    return dest;  
}
```

into `cs392_strncat.c`

If you want to test your function one-by-one, just replace the above code for that function with your code, and run the test commands.

You may find even the above code can pass some testcases. No worries. That's just randomness.