

CIS2107

Computer Systems & Low-Level Programming

Lab09. String Library

# Lab 9: String Library

→ Upload tar/zip file to Canvas

◆ Test on `cis-linux2` server !!!!

→ Comments at top of the file:

◆ Name, Date, Course

◆ Homework number (Lab 8 String Library)

◆ Statement of problem

**Warning !!**

**Cannot use `string.h`**

**library functions!**

**Please name your driver  
program `strtester.c`**

# Requirements

**ORGANIZATION:** create a directory

- **.c files:** one for each function in the library **AND** a test program [ 24 TOTAL ]
- **.h file:** contains declarations for each file, #included in each of the c files [ ONE TOTAL ]
- **.a file:** library file [ ONE TOTAL ]

# Important Commands

//compile all c files without linking - creates your executables!

```
gcc -c *.c
```

//create library file

```
ar rcs libstr2107.a *.o
```

//compile test program and link it with library

```
gcc -o strtester strtester.c -L. -lstr2107
```

```
./strtester
```

NOTE: libstr2107 and -lstr2107 are intentionally different

# Example Directory

Name	Size	Changed	Rights	Owner
_MACOSX		12/11/2018 10:39:57 AM	rw-r--r--	tuf87933
all_letters.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
all_letters.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
capitalize.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
capitalize.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
dedup.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
dedup.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
diff.c	1 KB	10/27/2018 11:46:03 AM	rw-r--r--	tuf87933
diff.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
ends_with_ignore_cas...	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
ends_with_ignore_cas...	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
find.c	2 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
find.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
is_empty.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
is_empty.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
len_diff.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
len_diff.h	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
len_diff.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
libstr2107.a	52 KB	10/27/2018 11:57:34 AM	rw-r--r--	tuf87933
num_in_range.c	1 KB	10/27/2018 11:46:03 AM	rw-r--r--	tuf87933
num_in_range.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
pad.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
pad.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
ptr_to.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
ptr_to.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
repeat.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
repeat.o	2 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
replace.c	2 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933
replace.o	3 KB	10/27/2018 11:56:57 AM	rw-r--r--	tuf87933
rm_empties.c	1 KB	10/27/2018 11:46:04 AM	rw-r--r--	tuf87933

hidden 0 B of 144 KB in 0 of 53

SFTP-3 0:00:50

# Example header file

```
/home/TU/tuf87933/CIS2107_Fall/Beck/Assignment8/string.h - tuf87933@cis-linux2.temple.edu - Editor - WinSCP

#ifndef string_h
#define string_h

int all_letters(char *s);
int num_in_range(char *s1, char b, char t);
int diff(char *s1, char *s2);
void shorten(char *s, int new_len);
int len_diff(char *s1, char *s2);
void rm_left_space(char *s);
void rm_right_space(char *s);
void rm_space(char *s);
int find(char *h, char *n);
char *ptr_to(char *h, char *n);
int is_empty(char *s);
char *str_zip(char *s1, char *s2);
void capitalize(char *s);
int strcmp_ign_case(char *s1, char *s2);
void take_last(char *s, int n);
char *dedup(char *s);
char *pad(char *s, int d);
int ends_with_ignore_case(char *s, char *suff);
char *repeat(char *s, int x, char sep);
char *replace(char *s, char *pat, char *rep);
char *str_connect(char **strs, int n, char c);
void rm_emptyies(char **words);
char **str_chop_all(char *s, char c);

#endif /* string_h */
```



# Example driver program

```
#include "string.h"
#include <stdio.h>
#include <ctype.h>

int main(void){

    puts("Test for all_letters");
    char *s = "hello";
    char *s1 = "sir";
    int all_letters_ = all_letters(s);
    if (all_letters_) {
        printf("String:%s\n", s);
        puts("All characters in string were letters\n");
    }else {
        puts("All characters in string were not letters\n");
    }

    puts("Test for num_in_range");
    int num_in_range_ = num_in_range(s, 'a', 'h');
    printf("The String:%s\n", s);
    printf("The range %c-%c", 'a', 'h');
    printf("%s %d %s\n", "There are", num_in_range_, "letters in the range");

    puts("Test for diff");
    int diff_ = diff(s, s1);
    printf("The Strings: %s, %s\n", s, s1);
    printf("%s %d %s\n", "The strings differ in", diff_, "ways");

    puts("Test for shorten");
    puts("Before:");
    printf("%s\n", s);
    puts("After:");
    char word[] = "hello";
    shorten(word, 3);
    printf("%s\n", word);

    puts("Test for len_diff");
    int len_diff_ = len_diff(s, s1);
    printf("The Strings: %s, %s\n", s, s1);
    printf("%s %d\n", "The strings differ in length by", len_diff_);

    puts("Test for rm left space");
```

# Example c file

```
/home/TU/tuf87933/CIS2107_Fall/Beck/MeghanBeckAssignment8/find.c - tuf87933@cis-linux2.temple.edu - Editor - WinSCP
#include "string.h"
#include <stdio.h>

int find(char *h, char *n) {
    int len_n, len_h;
    for(len_n = 0; *(n+len_n) != '\0'; len_n++);
    for(len_h = 0; *(h+len_h) != '\0'; len_h++);

    if (len_n > len_h) {
        return -1;
    }
    int count_n;
    int test[len_n];
    for (count_n = 0; count_n < len_n; count_n++) {
        test[count_n] = -1;
    }

    int h_counter, n_counter;
    h_counter = 0;
    n_counter = 0;

    int toRet = 0;

    while(*(h+h_counter) != '\0') {
        if (*(h+h_counter) == *n) {
            toRet = h_counter;
            while((*(n+n_counter) != '\0') && (*(h+h_counter) != '\0')) {
                if(*(n+n_counter) == *(h+h_counter)) {
                    test[n_counter] = 1;
                }
                else {
                    break;
                }
                n_counter++;
                h_counter++;
            }
            int sum_n = 0;
            for (n_counter = 0; n_counter < len_n; n_counter++) {
                sum_n += test[n_counter];
            }
        }
        h_counter++;
    }

    return toRet;
}
```

# 1. `int all_letters(char *s)`

➤ Input: "Hello World"

➤ Returns: 1

➤ Input: "He11o World"

➤ Returns: 0

2. `num_in_range(char *s1, char b, char t)`

➤ Input: "Yellow" 'f' 'm'

➤ Returns: 2

### 3. `diff(char *s1, char *s2)`

➤ Input: "Book" "Back"

➤ Returns: 2

## 4. void shorten(char \*s, int new\_len)

➤ Input: "Hello World" 5

➤ Returns: "Hello"

➤ Input: "Hello World" 20

➤ Returns: "Hello World"

5. `int len_diff(char *s1, char *s2)`

➤ Input: "Philadelphia" "Hello"

➤ Returns: 7

5. `void rm_left_space(char *s)`

➤ Input: " Hello"

➤ Returns: "Hello"



6. `void rm_right_space(char *s)`

➤ Input: `"Hello "`

➤ Returns: `"Hello"`

## 8. `void rm_space(char *s)`

➤ Input: `" Hello "`

➤ Returns: `"Hello"`

**9. int find(char \*h, char \*n)**

➤ Input: "Hello " "l"

➤ Returns: 2

➤ Input: "Hello" "q"

➤ Returns: -1

**10. char \*ptr\_to(char \*h, char \*n)**

➤ Input: "Hello " "l"

➤ Returns: pointer to l

➤ Input: "Hello" "q"

➤ Returns: NULL

## 11. `is_empty(char *s)`

➤ Input: `" "`

➤ Returns: `1`

➤ Input: `"Hello"`

➤ Returns: `0`

**12. str\_zip(char \*s1, char \*s2)**

- Input: "Temple" "Hello"
  - Returns: "Theemlp1loe"

## 13. `void capitalize(char *s)`

- Input: `"hello world"`
  - Returns: `"Hello World"`

**14. int strcmp\_ign\_case(char \*s1, char \*s2)**

➤ Input: "hello" "goodbye"

➤ Returns: 1

➤ Input: "Hello" "hello"

➤ Returns: 0



**15. void take\_last(char \*s, int n)**

➤ Input: "hello" 3

➤ Returns: "llo"

➤ Input: "hello" 6

➤ Returns: "hello"

## 16. dedup(char \*s)

- Input: "hello"
- Returns: "helo"

## 17. pad(char \*s, int d)

- Input: "hello" 6
  - Returns: "hello "
- Input: "hello" 5
  - Returns: "hello"

18. `ends_with_ignore_case(char *s, char *suff)`

➤ Input: "Coding" "ing"

➤ Returns: 1

➤ Input: "Coding" "ed"

➤ Returns: 0

19. `char *repeat(char *s, int x, char sep)`

➤ Input: `"hello" 3 '-'`

➤ Returns: `"hello-hello-hello"`

20. `char *replace(char *s, char *pat, char *rep)`

➤ Input: "Steph is the X" "X" "best"

➤ Returns: "Steph is the best"

21. `char *str_connect(char **strs, int n, char c)`

➤ **Input:** "Hello world Hello world" 4 '-'

➤ **Returns:** "Hello-world-Hello-world"

## 22. void rm\_empties(char \*\*words)

- Input: "Hello" "World" " " " " "Steph"
  - Returns: Hello World Steph



23. `char **str_chop_all(char *s, char c)`

➤ Input: "Hello/world/hello/world" '/'

➤ Returns: Hello, world, hello, world

# Checklist

- Do I have **23 separate c files** that each contain a single function?
- Do I have **a c file** that is the **driver program**?
- Do I have **a header file**?
- Does my program compile and run on the **cis-linux2** server?