**MCS 253**

**HW 2**

[Read this first on what to submit for a homework.](https://docs.google.com/document/d/1Gfy08swH0b0HSJ4kXGJLBvraCGz8ywxvmzDQhTQv3QM/edit?usp=sharing)

1. (50 points) Write 10 of the library functions defined by string.h for working with c-strings as documented here <http://www.tutorialspoint.com/c_standard_library/string_h.htm>. <http://www.codingunit.com/c-tutorial-strings-and-string-library-functions> You are to write strlen, strcpy, strcat, strcmp, strncmp, strchr, strpbrk, strstr, strspn, strtok. Write a program to test your implementations of each of your string/memory functions from string.h. Write at least good three test cases per string function. main can call each of the 10 test function (so it may be longer than 5 lines). Write one test function for each string function that has at least three test cases. These test function may also exceed the 5 line limit. I started the main function for you so use this as a start and follow my example for how to grow this for all 10 functions.

#include <string.h>

#include <stdio.h>

#define require(e) if (!(e)) fprintf(stderr, "FAILED line %d %s: %s %s\n", \_\_LINE\_\_, \_\_FILE\_\_, \_\_func\_\_, #e)

void test\_strlen()

{

require(strlen("") == 0);

require(strlen("hello") == 5);

}

void test\_strcmp()

{

char \*r = "hello", \*s = "hello", \*t = "he";

require(strcmp(r,s) == 0);

require(strcmp(t,r) < 0);

require(strcmp(s,t) > 0);

}

int main()

{

test\_strlen();

test\_strcmp();

/\* add more tests \*/

return 0;

}

1. (50 points) Write a program, called parseC, to parse a C program into tokens. Pay special attention to identifiers (start with an alpha or an underscore), numbers (start with a digit), string literals (start with a double quote, may contain escapes like \n or \t or \”), character literals (start with a single quote), operators (+-\*/=&|^%!?><~), and other special symbols (#[]{};:). You should recognize and discard program comments (/\* to \*/). Input should be read from stdin and output should be written to stdout with one token per line. Do not print any other output other than the tokes as we will use an automated script to test that your program functions correctly. Use your solution to homework 1 as test input for this program. Use Bash I/O redirection to test this program on your homework 1 solution (assuming it is in a file musicLibrary.c) like this: (show your program works on this same program below)

$ parseC < musicLibrary.cpp > tokens

**Sample input**

#include <stdio.h>

int main(int argc, char \*argv[]) /\* here is a comment \*/

{

char my\_char = ‘A’;

for (int i=0; i<1024; ++i)

printf(“\”Hello\” she said.\n”);

char hello = ‘B’;

}

**Sample output**

#

include

<

stdio

.

h

>

int

main

(

int

argc

,

char

\*

argv

[

]

)

{

char

my\_char

=

‘A’

;

for

(

int

i

=

0

;

i

<

1024

;

++

i

)

printf

(

“\”Hello\” she said.\n”

)

;

}