

Hack

Strings & String Processing

Computer Science I

Department of Computer Science & Engineering
University of Nebraska–Lincoln

Introduction

Hack session activities are small weekly programming assignments intended to get you started on full programming assignments. You may complete the hack on your own, but you are *highly encouraged* to work with another student and form a hack pair. Groups larger than 2 are not allowed. However, you may discuss the problems *at a high level* with other students or groups. You may not share code directly.

If you choose to form a Hack Pair, you *must*:

1. Both join a hack pair on Canvas (go to People then Hack Pairs)
2. You must both work on the hack equally; it must be an equal effort by both partners. Do not undermine your partner's learning opportunity and do not undermine your own by allowing one partner to do all the work.
3. Turn in only one copy of the code under the individual whose last name comes first (with respect to Canvas).

You are graded based on style, documentation, design and correctness. For detail, see the general course rubric.

Category	Point Value
Style	2
Documentation	2
Design	5
Correctness	16
Total	25

Table 1: Rubric

Correctness:

- 4 for student tester; it needs $3 \times 5 = 15$ test cases
- 3 total for replaceChar/replaceCharCopy tests
- 3 total for removeChar/removeCharCopy tests
- 4 total for lengthSplit
- 2 for error checking suite

Exercises

To get more practice working with strings, you will write several functions that involve operations on strings. In particular, implement the following functions with the described behavior. You *must* use the given signatures.

1. Write a function that replaces instances of a given character with a different character in a string.

```
void replaceChar(char *s, char oldChar, char newChar);
```

Which will replace any instance of the character stored in `oldChar` with the character stored in `newChar` in the string `s`.

2. Write a function that takes a string and creates a new copy of it but with instances of a given character replaced with a different character.

```
char * replaceCharCopy(const char *s, char oldChar, char newChar);
```

3. Write a function that takes a string and removes all instances of a certain character from it.

```
void removeChar(char *s, char c);
```

When removing characters, all subsequent characters should be shifted down. Take care that you handle the null terminating character properly.

4. Write a function that takes a string and creates a new copy of it but with all instances of a specified character removed from it.

```
char * removeCharCopy(const char *s, char c);
```

Take care that the new copy does not waste memory.

5. Write a function that takes a string and splits it up to an *array* of strings. The split will be length-based: the function will also take an integer n and will split the given string up into strings of length n . It is possible that the last string will not be of length n . You will not need to communicate how large the resulting array is as the calling function knows the string length and n .

`char **lengthSplit(const char *s, int n);` For example, if we pass

"Hello World, how are you?" with $n = 3$ then it should return an array of size 9 containing the strings "Hel", "lo ", "Wor", "ld,", " ho", "w a", "re ", "you", "?"

Instructions

- Place all your function prototypes into a file named `string_utils.h` and their definitions in a file named `string_utils.c`.
- In addition, create a main test driver program called `stringTester.c` that demonstrates at least 3 cases per function to verify their output. Hand in your tester.
- You are encouraged to collaborate any number of students before, during, and after your scheduled hack session.
- You may (in fact are encouraged) to define any additional “helper” functions that may help you.
- Include the name(s) of everyone who worked together on this activity in your source file’s header.
- Turn in all of your files via webhandin, making sure that it runs and executes correctly in the webgrader. Each individual student will need to hand in their own copy and will receive their own individual grade.