

CSCI 247 Computer Systems I Summer 2017 Laboratory Exercise 1

Objectives

1. Gain some familiarity with the C programming language.
2. Practice compiling and running C programs.
3. Gain some experience with characters integers and strings in C.

Submitting Your Work

Submit your C program files as via the **Lab Exercise 1 Submission** item on the course web site. You must submit your program by 2:00pm on the Monday following your scheduled lab session.

Introduction

Your task for this lab is to complete a C program that performs substring substitution on lines of text read from `stdin`. The program ensures that the substring to be replaced and the substring it is to be replaced with are of exactly the same length. For example, if the file `data.txt` contains

```
This is test data to be used for Lab exercise 1.  
This file contains some strings and the idea is to  
replace substrings by other substrings.  
stuttering statistical storms and tempests!
```

If we run the program to replace every occurrence of "st" by "xx", redirecting `stdin` to come from this file:

```
./Lab1 "st" "xx" < data.txt
```

The output should be:

```
This is texx data to be used for Lab exercise 1.  
This file contains some xxrings and the idea is to  
replace subxxrings by other subxxrings.  
xxuttering xxatixxical xxorms and tempexxs!
```

For this program you start with an existing, but incomplete program `Lab1.c`, which is available in the `Lab1` folder on the course web site. The existing program uses two functions `find()` and `replace()` which are

declared, but not defined in `Lab1.c`. Your task is to complete the definitions of those functions. Here are the declarations of the functions:

```
int find(char *substr,
         char *str)
```

This function returns the start index of the string `substr` in the string `str`. If `substr` cannot be found in `str`, `find()` must return `-1`.

For example,
`find("st", "substitute substring");`
will return the value 3.

Note: There are functions in the C library, which can do this task, however, you are **NOT** permitted to use them. You may only use the library function `strlen()`, which will tell you the number of characters in a string. In order to use that function you should include the header file `string.h`.

```
void replace(char *str,
             int pos,
             char *newstr)
```

This function replaces the characters in the string `str`, starting at index `pos`, by the string `newstr`.

For example,
`char line[] = "this is an example";`
`replace(line, 2, "at");`
will change `line` to `"that is an example"`.

After modifying `Lab1.c`, compile it to produce the executable file `Lab1`:

```
gcc -Wall -o Lab1 Lab1.c
```

Then you can run the program, as shown in the example above.

Submit your file via the course web site

Submit the file `Lab1.c`.