Tutorial 4
Strings

Learning Objectives

After this tutorial, you will be able to:

- Manipulate strings by accessing and modifying their component characters
- Reference man pages and documentation to solve problems

Tutorial

In order to receive full marks, you must complete everything in the “Tutorial” section. Part marks may be provided if good progress was made toward completing the tutorial, at TA discretion.

1. Download the file `T04.tar.gz` from the tutorial module on Brightspace.

   This file is both archived (tar) and compressed (gzip). Use the `man` / `help` pages to see how to extract the files in this file. Using `gzip` to compress a tar file is a common way to compress files in Unix/Linux environments, and is essentially equivalent to zipping files.

   Background: A Caesar shift cipher is an encryption cipher where each letter in the plaintext is replaced with a letter a fixed number of positions away in the alphabet. The number of positions that the letters are shifted forms a key for this cipher. For example, with a shift of 5 to the right, ‘A’ becomes ‘F’, ‘B’ becomes ‘G’, ‘Y’ becomes ‘D’ and so on. You can learn more about the Caesar shift cipher here. Note that only letters are affected by the cipher, and non-letter characters remain unchanged.

2. Finish writing the functions `encryptCaesar()` and `decryptCaesar()` so that they encrypt and decrypt a Caesar shift cipher with a shift of `key` to the right.

3. The string `cipher` has been encrypted using a Caesar shift cipher, but the key is unknown! Determine what the key is, and decrypt the message. **Hint:** Are there really *that* many possible keys?

Exercises

These exercises are optional, but are provided to give you some challenging problems to solve that will have you working extensively with strings and arrays. There are multiple plausible solutions. Play around, and see if you can make progress toward them. **Tip:** Special characters make things tough. Writing a helper function might apply across multiple exercises.

1. Write a function `camelCase(char *str)` which removes all spaces from the in/out parameter `str` and capitalizes the first letter of every word (except the first) in `str`.
2. Write a function `slugify(char *str)` which removes all non-letter, non-number characters from `str`, and replaces any whitespace between the words with a single hyphen to separate words. All letters should be put lower case. For example:

   ```
   char plain[] = “ - Friends, Romans, countrymen,  lend me your ears?”;
   slugify(plain);
   printf(“%s\n”, plain);
   ```

   Should output:

   ```
   friends-romans-countrymen-lend-me-your-ears
   ```

(Exercises continue on the next page)

Exercises (cont.)

The Caesar shift cipher is used in more complex ciphers, such as the Vigenère cipher, whose description is given here. Note that instead of consulting a Vigenère table, when encrypting with a letter from the key, it is equivalent to apply a Caesar shift to the right with the 'value' of the key letter, where 'A' = 0, 'B' = 1, … , Z = '25'. For example, to encrypt with 'F', apply a Caesar shift of 5 to the right.

3. Using the functions that you wrote for the Caesar shift cipher, write the functions `encryptVigenere(char *plaintext, char *key)` and `decryptVigenere(char *ciphertext, char *key)`, which encrypts and decrypts a Vigenère cipher with a key given by the string `key`.

Saving and Submission

In-Person Attendance: Submission is optional, as you can be checked off in class.
Asynchronous: If you are completing the work on your own at home, make sure to follow the submission requirements.

1. Simply submit your `T04.c` file with the questions completed. Make sure to remove any debugging code that you included to simplify the grading process.
    a. If you are more familiar with packaging and submitting tar files, you may do this as well
    b. Make sure not to submit a `T04.c` file that is only the base, provided code.
2. For **full marks** you should have completed all problems in the Tutorial section (and they should be seen upon running the program). The exercises are optional, but highly recommended for the extra challenge and exploration of these concepts.
3. For **part-marks** you should have attempted to complete most of the tutorial. Grades for part-marks will be at TA discretion.