

CS 100 Project Four – Fall 2017

Project Overview: This project reads a file of text and generates a cipher of the file. A cipher is simply a secret or disguised way of writing the text. We disguise the text by shifting it to the right (or left) and also incrementing (or decrementing) the letters and digits in the text.

The user supplies, on the command line, an input file and a combination of letters (**I** or **D** or **L** or **R**) to create a unique cipher that is applied to the input file. Your program will read the file word by word. A word, also known as a token, is defined as a string bounded by white spaces but containing no white spaces. It applies the following algorithm to each word read from the input file.

- **I** – increment all the letters and digits in the word. That is, an **a** becomes a **b**, an **x** becomes a **y**, a **1** becomes a **2**, and so on. If you have a **z** or a **Z** or a **9**, then these wrap around and become an **a** or **A** or **0**. If a character in the word is not a letter or a digit, it is kept unchanged.
- **D** – decrement all letters and digits in the word. That is, a **b** becomes an **a**, an **N** becomes an **M**, a **5** becomes a **4**, and so on. If you have an **a** or **A** or **0**, then these wrap around and become a **z** or **Z** or **9**. Again if a character in the word is not a letter or a digit, it is kept unchanged.
- **L** – shift all the characters in the word to the left one position. The character at position 0 (the start of the word) becomes the new last character in the word.
- **R** – shift all the characters in the word right by one position. The last character in the word becomes the new first character (the start of the word).

Your program gets the name of the input file and the sequence of cipher commands from the command line. It writes the cipher to standard output (one token/word per line). As an example of how the program works, consider the input file shown below (named **data**)

The University of Alabama *1831* Crimson Tide!			
./a.out data L	./a.out data I	./a.out data LI	./a.out data LIDR
heT niversityU fo labamaA 1831** rimsonC ide!T	Uif Vojwfstjuz pg Bmbcbnb *2942* Dsjntpo Ujef!	ifU ojwfstjuzV gp mbcbnbB 2942** sjntpoD jef!U	The University of Alabama *1831* Crimson Tide!

Two quick comments on this program:

- You **must** use at least four functions in your program and place their signatures at the beginning of the file. One obvious approach is to have a function for each of the four cipher techniques.
- Your program **must** print an error message and exit if the program fails to open the file specified for reading, or if any of the “encryption” commands are not valid – must be one of **I** or **D** or **L** or **R**.

What You Need To Do

- Create a directory **project4** on your machine. In that directory, create a file named **cipher.c**
- In **cipher.c**, write the code needed to implement this program. Make sure that you include a header block of comments with your name and a brief overview of the program.
- To submit your project, first bundle your **project4** directory into a single (compressed) zip file.
- Once you have a compressed zip file that contains your **project4** code, submit that file to Blackboard.

Project 3 is due at 5:00pm on Friday, October 20. Late projects are not accepted.