17.7 Project 4: schwifty

Name this program `schwifty.c` - This program reads a text file and makes it schwifty. The user supplies the filename to schwift and a string containing a sequence of the following characters to determine the schwiftiness via command line arguments:

- `L` - Left shift each character in a word:
  - hello --> elloh
- `R` - Right shift each character in a word:
  - elloh --> hello
- `I` - Shift the letters' and digits' value "up" by one (wraps if necessary) and keeps other characters as the same:
  - a --> b
  - Y --> Z
  - 2 --> 3
  - z --> a (wrap)
  - 9 --> 0 (wrap)
  - [--> [(same: not letter or digit)
- `D` - Shift the letters' and digits' value "down" by one (wraps if necessary) and leaves other characters as the same:
  - b --> a
  - Z --> Y
  - 3 --> 2
  - a --> z (wrap)
  - 0 --> 9 (wrap)
  - [--> [(same - not letter or digit)

Example execution

`ex.txt`	`./a.out ex.txt R`	`./a.out ex.txt D`	`./a.out ex.txt RD`	`./a.out ex.txt RDIL`
Oh, yeah! You gotta get schwifty. *empty line*	,Oh !yeah uYo agott tge .schwifty *empty line*	Ng, xdzg! Xnt fnssz fds rbgvhesx. *empty line*	,Ng !xdzg tXn zfnss sfd .rbgvhesx *empty line*	Oh, yeah! You gotta get schwifty. *empty line*

ex.txt	./a.out ex.txt R	./a.out ex.txt D	./a.out ex.txt RD	./a.out ex.txt RDIL
	Right Shift	Decrement	Right Shift, then Decrement	Right Shift, Decrement, Increment, Left Shift

Notes

- A word is a group of characters surrounded by whitespace.
- You can assume no word's length will be greater than `100`.
- Each schwiftified word is written to `stdout` on a newline as shown in the examples.
- Implement the following four functions in your program that each performs the associated schwift on a word. Your functions will be unit tested, so implement the signatures exactly as shown. A starting code template is downloadable below.
    1. `void left(char word[]);`
    2. `void right(char word[]);`
    3. `void inc(char word[]);`
    4. `void dec(char word[]);`
- Error messages:
    1. If there is not exactly `3` command line arguments, then print out the following error message: "`Invalid number of arguments`".
    2. If the input file cannot be opened, then print out the following error message: "`Could not open file 'filename'`" where filename is the name of the file (surrounded by single quotes).
    3. If one of the schwifties is invalid (not `L`, `R`, `I`, or `D`), then print out the following error message and do not print any words: "`You threw off my schwiftiness with schwifty X!`" where `X` is the leftmost invalid schwifty encountered in `argv[2]`.
    - If errors `2.` and `3.` both occur at the same time, only print `2.`'s error message.

LAB ACTIVITY | 17.7.1: Project 4: schwifty | 100 / 100

Submission Instructions

Downloadable files

`schwifty.c` **Download**

Compile command

```
gcc schwifty.c -Wall -o a.out -lm
```

We will use this command to compile your code

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

schwifty.c

Drag file here
or
Choose on hard drive.

Submit for grading

1 submission left

Latest submission - 6:35 PM on 07/12/20 — Submission passed all tests ✓ — Total score: 100 / 100

☐ Only show failing tests

Download this submission

1: Error: + arg count ^ — 2 / 2

Run command `./a.out input1.txt L extra`

Output results hidden by your instructor

2: Error: - arg count ^ — 2 / 2

Run command `./a.out not_enough`

Output results hidden by your instructor

3: Error: bad file ^ — 2 / 2

Run command `./a.out does_not_exist LIRD`

Output results hidden by your instructor

4: Error: bad file and bad schwifties ^ — 2 / 2

Run command `./a.out does_not_exist LIRDB`

Output results hidden by your instructor

5: Error: bad schwifties 2 / 2

Run command `./a.out input1.txt LIRDB`

Output results hidden by your instructor

6: L 3 / 3

Run command `./a.out input1.txt L`

Output results hidden by your instructor

7: I 3 / 3

Run command `./a.out input1.txt I`

Output results hidden by your instructor

8: R 3 / 3

Run command `./a.out input1.txt R`

Output results hidden by your instructor

9: D 3 / 3

Run command `./a.out input1.txt D`

Output results hidden by your instructor

10: LI 5 / 5

Run command `./a.out input1.txt LI`

Output results hidden by your instructor

11: LIDR 8 / 8

Run command `./a.out input1.txt LIRD`

Output results hidden by your instructor

12: Schwifty Manifest in Its Purest Form 10 / 10

Run command

```
./a.out input1.txt
LIRDLRLLLLRLLRRRRRRRRRRDDDDDDDDDDDDDDIDIIIDDIIIIIIIII
```

Output results hidden by your instructor

13: left() 10 / 10

14: right() 10 / 10

15: dec() - wrap 9 / 9

Test feedback

```
res: zabcdefghijklmnopqrstuvwxyZABCDEFGH
ans: zabcdefghijklmnopqrstuvwxyZABCDEFGH
```

16: dec() - no wrap 6 / 6

17: inc() - wrap 9 / 9

18: inc() - no wrap 6 / 6

19: LIRD on a single character word 5 / 5

Previous submissions

6:33 PM on 7/12/20	98 / 100	View

Trouble with lab?