---

ROT13.c

Description

---

The purpose of this program is to let you practice string and array manipulation as well as bit-wise operation in C.

In this project, you are asked to implement a function rather than a whole program. The main body of the program is provided by the instructor. A makefile is also provided which allows you to compile the demo program as well as your implementation.

The program in this project is about ROT13 encoding.
ROT13 stands for "rotate by 13 places". A similar form of this also known as Caesar cipher. The algorithm is not really useful for provide security protection and is often cited in history of cryptography or as a canonical example of weak security.

Your program needs to provide the following function:

> For any input, convert any letters that are in the 'a'-'z' and 'A'-'Z' ranges with the rule of ROT13, but keep other characters outside these ranges unchanged.
>
> The program given to you have a two modes: encoding and decoding. (They are essentially the same function if you have not figured that out).

Files:

A main.c and rot13.o files are provided so that you can run the program. A Makefile is provided so that if you type "make demo", a demo program with rot13 encoding/decoding ability is generated.

A makefile and testcases are also given to you so that you can run test by yourself. To test the demo program, run a "make demo" followed by a "make test". Or run "make demo; make test" in one line.

To test your implementation, run a "make" to generate your version of project2 executables. If success, a new 'proj2' file is created, and you can run "make test" to test your implementation.

Since the rot13.o are generated on a virtual machine that is exactly like yours, you might not be able to test the demo program other than the virtual machine assigned to you.

Your task is to produce your version of rot13 implementation in file rot13lib.c .

Your implementation has to be usable by the main.c and you are NOT allowed to change the main.c and the makefile

SUBMIT REQUIREMENTS:

* You must submit at least the following files (i.e., all the files necessary to compile your program):

README.txt
main.c (please submit main file even you are NOT allow to change it)
rot13lib.c (your implementation of rot13)
Makefile (similar to main.cpp, please DO NOT change this file)

In the "README.txt", you should list your name and the command used for compiling your code. If your makefile does not work, TA will try to use the command in your "README.txt" compile your code.

The command format for executing your executable file should be
make; make test

If you are curious the inside of the execution, the proj2 program actually takes the following command line arguments. These details are hidden from you by the makefile setup.
proj2 e < "file-to-be-encoded"

To submit the files, you will need to copy these files (README.txt, main.c, rot13lib.c, Makefile) to your computer, and then upload them to ELC-new

---

Grading Policy (test cases will be release later)

grading calculation (test cases will be released later)

submit valid readme file 10 points

submit valid make file  10 points.
test case 1 20 points
test case 2 20 points
test case 3 20 points
test case 4 20 points

Note that assignment that can not be compiled will be graded ZERO.
---