
mobydick.c

Description

The purpose of this program is to introduce you to programming in C using strings and arrays. As usual make sure to modularize your program properly (use functions when appropriate) and include meaningful comments.

Check main.c to find the anagrams in the text Moby Dick by Herman Melville

An anagram is a string of letters that are scrambled or have transposed another word or (random) phrase. As an example, the string:

 Itsgrace
is an anagram for the phrase:

 C is great

Note that an anagram consists of letters and digits.

All spaces in the text are ignored.

All texts are NOT case sensitive.

Hence, your program needs to solve:

What is the phrase, or phrases, and where do they appear in the book that match the anagrams?

You should read the MobyDick text from standard input (i.e. use getchar).

The characters in the anagram (the original search string) may be upper or lower case.

The text is available at:

<http://www.textlibrary.com/download/moby-dic.txt>

already downloaded text is available on the class schedule web page.

NOTE 1:

To read the file using standard input, you redirect the file

"moby-dick.txt to standard in on the command line by using the operator "<". For example if your executable file is called "MobyDick.out" you execute the program as follows:

```
MobyDick.out < moby-dick.txt
```

NOTE 2:

The variables that need to be filled in can be found in main.c
Do not modify the contents of any file besides mobydick.c and mobydick.h (if needed)

EXAMPLE:

As an example, if you were to call the find_anagram(...) function on the provided text using the string "Llama ice helmets", you can expect the output to read:

```
Anagram 1 is: Call me Ishmael  
Anagram 1 was found on line: 15
```

SUBMIT REQUIREMENTS:

- * print the decoded anagram(s) and its location (i.e. the line number) in the text.
- * properly commented code. (e.g. with author info on the first line as a comment)

- * Submit ONLY the following files:

```
README.txt  
mobydick.c  
mobydick.h  
Makefile
```

Be sure the README.txt file contains your name.

You will need to submit a copy of your project to eLC before the deadline date. You will also need to keep a copy on your VM. Please follow these instructions. If you do not put your project in the correct folder with the correct name, you may receive a 0 for your grade.

- Navigate to your home directory: cd ~/

- Create a folder named "project3": `mkdir project3`
- Place all your source code, including a readme and a Makefile, in the project3 folder

GRADING POLICY:

Up to 10 BONUS POINTS for speed:

The top 5 fastest (most efficient) project will receive bonus (we will determine this by using the Unix time command when grading). Winner will receive an additional 10, 8, 6, 4, 2 points respectively.

Rubric

- submit valid readme file 10 points
- submit valid makefile 10 points.
- test case 1 20 points
- test case 2 20 points
- test case 3 20 points
- test case 4 20 points

Please note: The test cases that we use when grading your project will NOT be the same as the ones you are given in the provided main.c.

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