Word(s) Search [contributed by F. R. Salvador]

Enable2k is a publicly available list of English words. It is useful for creating programs such as word games that require searching a list that matches a specified pattern. For example, searching for all words (in Enable2k) that start with the letter ‘q’ and ends with ‘t’ results into 42 matches, including “quadrant”, “quiet” and less familiar words such as “qiviut” and “quaggiest”.

PRELIMINARY TASK (not graded yet): Open “ENABLE2K.txt” text file that comes together with this questionnaire. Familiarize yourself with the arrangement of the list by answering the following questions. Answers are in the footnote[1].

Q1: Are the words stored in lower case or upper case letters?
Q2: Are the words arranged in some order?
Q3: How many words are there all in all in the list?

ACTUAL GRADED TASK: Write 6 separate functions that return values as respective answers to the following questions:
Q1: Is the word program in the list? *(Note: this is a searching problem. Your function should return 1 to mean “Yes, the word is in the list; otherwise, it should return 0 which means, “No, the word is not in the list”).*
Q2: How many 10-letter words that start with the letter g are there in the list? *(for example: gatekeeper)*
Q3: How many words begins with the letter q and ends with the letter t? *(for example: fry)*
Q4: How many words are there that do not contain any vowel? *(for example: fry)*
Q5: How many words are there that start with the string unli? *(for example: unlimited)*
Q6 (Bonus Question): What is the longest word? *(Note: your function must search for this word)*

Note that those in red font and yellow background are just example values. They can be replaced with other values. This means, that in the function definitions, they are implemented as function parameters.

You are given the following accompanying files:

- **wordsearch.h** is a header file that contains the macro definitions and function prototypes of the functions that you need to define
- **WORDSEARCH-LASTNAME.c** is the skeleton file where you will encode your six function definitions
- **main.c** contains the main() function that you need to complete and you’ll need to use to test your five functions
- **PENABLE2K.txt** is a text file that contains the words; there are 173,528 words/strings in the text file.
- **ANSWER.txt** is a text file that contains the answers to the questions above except for Q6 which is NOT_REVEALED.

Make sure to rename the skeleton file with your own last name. For example, if your lastname is SANTOS, you should have a file named WORDSEARCH-SANTOS.c.

RUN YOUR PROGRAM WITH OUTPUT REDIRECTION: Run your exe file in the command line with input redirection. You can store the result of your program into a text file by output redirection, For example, if your last name is SANTOS, then you can run your executable program with output redirection as:

```
C:\CCPROG2> WORDSEARCH-SANTOS > SANTOS-RESULT.txt
```

The output of the exe file will be stored in SANTOS-RESULT.txt file. A correct program should produce the same set of values that are in ANSWER.txt using the same set of parameters in the examples for Q1 and Q5 above. There should only be one difference which is for Q6. The correct is not revealed for now… The challenge is for your to figure it out on your own.

TESTING & SCORING:

- Your program will be black box tested by your instructor with other parameter values.
- A program with a compilation error will AUTOMATICALLY result into a score of 0. Make sure that there are no syntax/compilation errors in your solution.
- Correct solutions for Q1 to Q6 will be given 10 points each. The score for an incorrect implementation of a required function is 0. For example, if only the solutions for Q1, Q2 and Q3 are correct, then the score will be 30/50.

DELIVERABLES:
Submit/upload two files via Canvas before the indicated deadline:

1. your WORDSEARCH-LASTNAME.c source file
2. your LASTNAME-RESULT.txt output text file

Don’t forget to change the filename to your own last name.

-- The End --

[1] A1: They are stored in lower case. A2: Yes, the words are arranged alphabetically. A3: There are 173,528 words.