Assignment #1

Computer Science I – COP 3502

Objectives

1. To review important C concepts: arrays, files, functions, and structures

Problem: Scrabble Practice (scrabble.c)

Scrabble is word game in which each player has seven letters with which to make the best possible word. Words are judged on a point value system: each letter is worth a specific number of points and each word is worth the sum of each letter's point.

For example, a player may be able to form the word CAT. C's are worth 3 points and both A's and T's are worth 1 point. So the word CAT would be worth 3+1+1 or 5 points total.

In this assignment you will be writing a program that allows the user to practice their scrabble skills. First, you must read in a prewritten dictionary from a text file. The simplest storage method for this is an array of strings.

Then, you will need to generate a set of random letters. All letters should be upper case. Recall that the ASCII values for upper case letters range from 65-90. You must ensure that the set of letters is valid – you cannot have seven A's in your set even if random chance produced the value 65 seven times.

In general, your set of letters must fit the normal distribution of Scrabble tiles. This distribution is below along with the point values of each letter.

When you have generated a set of seven random letters, show these to the users. Then, prompt the user with the following options:

1 – Enter Word
2 – Quit

If the user chooses to enter a new word, you must read in the word and ensure it is valid. The user's word is valid if it is in the dictionary and they have the letters to form that word. For valid words, you will print out the point value for that word. For invalid words, you will print out that the word is invalid. Then, prompt the user with the above options again and until they decide to quit.

When the user decides to quit, print their best word and its associated point value. If the user has not entered any words, do not print anything.

Letter	Point Value	Distribution
A	1	9
B	3	2
C	3	2
D	2	4
E	1	12
F	4	2
G	2	3
H	4	2
I	1	9
J	8	1
K	5	1
L	1	4
M	3	2
N	1	6
O	1	8
P	3	2
Q	10	1
R	1	6
S	1	4
T	1	6
U	1	4
V	4	2
W	4	2
X	8	1
Y	4	2
Z	10	1

File Specifications

1. The input file will be stored in dictionary.txt. You will not need to ask the user for the file.
2. The first line of the input file will be an integer, n.
3. The following n lines will each contain one word belonging to the dictionary.
4. All words (both in the dictionary and entered by the user) will be upper case.
5. The dictionary cannot have more than 10000 words in it.
6. Each word will have no more than 7 letters in it.

Program Specifications

1. The user will only enter number 1 or 2 for option selection.
2. You must write functions to complete the following tasks. The functions' parameters and prototypes will be up to you.
    a. Generate a set of seven random letters
    b. Determine if a user's word is valid
    c. Calculate the value of a valid word
3. Bonus points may be earned for using a function to read in the dictionary from the file

Sample Run

Below is a sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above.

In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Welcome to the Scrabble Practice Program!

Here are your letters: SHABCTC

What would you like to do?
1 – Enter Word
2 – Quit

1
Word:*CAT*

That word is worth 5 points.
What would you like to do?
1 – Enter Word
2 – Quit

1
Word: *CATCH*

That word is worth 12 points.
What would you like to do?
1 – Enter Word
2 – Quit

1
Word: *CAB*

That word is worth 7 points.
What would you like to do?
1 – Enter Word
2 – Quit

2

Your best word was CATCH worth 12 points.

Deliverables

One source files – *scrabble.c* – is to be submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Code::Blocks. Yourprogram should include a header comment with the following information: your name, course number, section number, assignment title, and date. Also, make sure youinclude comments throughout your code describing the major steps in solving the problem.

Grading Details

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.

3) Compatibility – You must submit C source files that can be compiled and executed in a standard C Development Environment. If your program does not compile, you will get a sizable deduction from your grade.