Computer Science I Program 0: Wordle Scoring

Please Check Webcourses for the Due Date

Please read the whole assignment before you start coding

Objective

Give practice with reading and writing to standard input in C.
Give practice with loops and conditionals in C.
Give practice with arrays and strings in C.

Background Story

You've been hired by the New York Times to support their burgeoning collection of daily games, the most popular of which is Wordle. Naturally, a helpful tool would be to give words a score simply based on the letters in the word. You've been asked to score each word in a list of words as follows:

1. First count up the number of occurrences of each letter, 'a' to 'z', in each of the word total. We'll call the score of a single letter the number of times that letter showed up in total over all the words.
2. Go back through each word in the list and add up the scores of each of the letters in the word. If a letter appears twice in the word, then that letter's score should be added twice, once for each occurrence in the word.
3. Print out each word in the list along with its score, in the original order the words appeared in the list.

Problem

Given a list of five letter words, score each of the words according to the method described above, and print out a list of the words in the order of the original list, with each word followed by a space and its score.

Input

The first line of input will contain a single positive integer, **n** ($2 \leq \mathbf{n} \leq 1000$), representing the number of words in the list. The next **n** lines of input will each contain a single five letter string of lowercase letters, representing the list in the original order.

Output

Output one line for each word in the input, in the order the words appeared in the input. Each of these lines should have the word, followed by a space, followed by the score of the word.

Sample Input	Sample Output
5 spare arose adieu thine raise	spare 16 arose 16 adieu 14 thine 11 raise 18
11 betty again canal mamma sauna radar carat plaza madam aroma gamma	betty 9 again 46 canal 47 mamma 64 sauna 45 radar 50 carat 49 plaza 44 madam 58 aroma 53 gamma 58

First Sample Explanation

In the first sample case, here is how often each letter appears: 'a' = 4, 'd' = 1, 'e' = 5, 'h' = 1, 'i' = 3, 'n'=1, 'o' = 1, 'p' = 1, 'r' = 3, 's' = 3, 't' = 1, 'u' = 1.

It follows that the score for spare = 3 + 1 + 4 + 3 + 5 = 16, score for arose = 4 + 3 + 1 + 3 + 5 = 16,
score for adieu = 4 + 1 + 3 + 5 + 1 = 14, score for thine = 1 + 1 + 3 + 1 + 5 = 11,
score for raise = 3 + 4 + 3 + 3 + 5 = 18

Implementation Requirements/Run Time Requirements

1. Please statically allocate your arrays to be big enough to read in the maximum input size of arrays as this assignment is meant to review some aspects of COP 3223 and not meant to contain new material from COP 3502.

2. For full credit, your program should have **a minimum of three auxiliary functions** that are different from the main function. Your function prototypes should be included after your #includes before main. The functions themselves should be located after main in your program.

3. For full credit, you must declare constants for the maximum array size, the maximum size of a word and the alphabet size.

4. Your code must compile and execute on the Eustis system. The C compiler on this system is the one that the graders will be using to grade/evaluate your submissions.

Deliverables

1. Please submit a single source file, `wordlescoring.c`, via Webcourses.