Computer Science I Program 2: Spelling Bee (Recursion)

Please Check Webcourses for the Due Date

Please read the whole assignment before you start coding

Objective

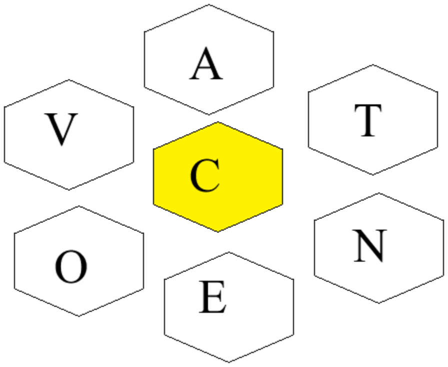
Give practice with recursion, specifically both the permutation algorithm and binary search algorithm

Give practice with functions in C.

Give practice with creating a design for a program without a given list of functions or structs.

Spelling Bee Problem - Modified

In the original NY Times Spelling Bee game, the player is presented with a hexagon of letters (6 letters on the outside with one center letter), similar to the picture below (yes my skills in MS Paint suck...):



In the original game, the goal is come up with as many words that are four letters or longer that use only the letters shown in the picture (repeats allowed) that have at least one copy of the middle letter, in this case, C.

In our modified version, we won't allow for repeated letters.

Problem

Given the seven letters for a spelling bee puzzle, with the fourth letter indicating the one required letter, as well as a sorted dictionary of valid words (all in between 4 and 7 letters long), determine all the valid words that can be formed according to the rules of the game, in alphabetical order.

Input

The first line of input contains a single string, ***s***, of 7 distinct lowercase letters, the fourth of which denotes the middle letter for the spelling bee puzzle.

The second line contains a single positive integer, ***n*** ($1 \le n \le 10^5$), indicating the number of words in the dictionary. The following ***n*** lines will each contain a string of lowercase letters, indicating valid words. Each of these strings will contain in between 4 and 7 letters, inclusive, and the strings will be sorted in alphabetical order.

Output

Output each valid word that can be formed by the letters in the puzzle using the middle letter, with one word per line, in alphabetical order.

Sample Input	Sample Output
votcane 10 able breakup cable cane cave cone tent vocate vone vote	cane cave cone vocate
rstlnea 24 able arsteln back cans deal forth gamma last lean learn nale nelarst rest restan smart staple star stars start tars trale trade zebra zoos	arsteln last lean learn nale nelarst trale

Implementation Requirements/Run Time Requirements

Note: This problem and the regular Spelling Bee problem can easily be solved with Intro to C methods. If you solve it this conventional way, you won't get any creditbecause the purpose of this assignment is to practice recursion, specifically both the permutation algorithm and the binary search algorithm. The permutation algorithm will have to be non-trivially modified while the binary search can be written nearly identical to how it was shown in class.

Thus, please follow these requirements, for the sake of practice the desired algorithms.

1. **You must sort the 7 letters in alphabetical order**, so that when you run the permutation algorithm, it naturally tries each possible arrangement in alphabetical order. Since we haven't taught sorting yet, either come up with an idea on your own or look one up. It's only 7 distinct letters, so there are quite a few ideas that will work.

2. The permutation algorithm from class must be used. The key modification is that you must adjust the algorithm to allow for words that don't use all of the letters while making sure the middle letter is used.

3. You **may use** global variables to clean up your code so it's easier to read. Please use them sparingly. (Here are the ones recommended: number of words in the dictionary and the dictionary itself. It would also be okay if the string storing the 7 letters of the puzzle is global.

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.

5. The intended solution to the problem uses 11 functions other than main. While you don't have to have exactly that many, it is expected, that if you design your program well that you'll have close to that many functions.

Note: Your instructor may choose to provide a set of function prototypes for you if they wish.

Deliverables

1. Please submit a source file `spellingbee.c`, via Webcourses, for your solution to the problem.