Homework #4
Due Wednesday, April 27, 11:59 PM
40 points total

Write your programs in C using the environment of your choice. The last program should be in C++. Submit your code via Blackboard as a zip file. See the posted documentation guidelines - for this assignment some problems will have points assigned to documentation. Also please give your files meaningful file names (e.g. problem1.c or gameshow.c, etc.) instead of the default such as source.c.

1) (10 points) Vowely Words

What English word has the most consecutive vowels? To answer this question of paramount importance, write a program that reads in this file that consists of 87337 English words: words.txt. For each word, count the number of consecutive vowels (a, e, i, o, or u) and output the word with the most consecutive vowels. If there is a tie, you only need output one of the words with the most consecutive vowels.

2) (5 points) Recursion Warmup - Binary Search

Read the words.txt file from problem 1 into an array so that each word is stored in the array (it can be a 2D array of char). Then, implement the recursive binary search algorithm (it is given in 10.3 and a version using an array of ints is in the lecture notes) so that the program allows you to input a word and then outputs whether or not the entered word is in the word list by using the binary searcch algorithm. The recursive algorithm should not use any global variables.

3) (5 points) Recursion Warmup - String Pairs

Let's say that a "pair" in a string is two instances of a char separated by a char. So in "AxA" the A's make a pair. Pair's can overlap, so "AxAxA" contains 3 pairs -- 2 for A and 1 for x. Complete the function below to recursively compute the number of pairs in the given string:

```
int countPairs(char *str)
{

}
```

Test your function with the following strings.

String	Correct # of pairs
"axa"	1
"axax"	2
"axbx"	1
"hi"	0
"hihih"	3
"ihihhh"	3
""	0
"a"	0
"aa"	0
"aaa"	1

The recursive algorithm should not use any global variables.

4) (10 points) Recursion - Dropoff Words

Let's define a "dropoff word" as one that can lose any one of its letters, and the remaining letters still makes up a valid word. This process repeats until there is only one letter left. If there is an invalid word along the way then it's not a dropoff word. Here are some examples:

- clink → link → ink → in → i
- strippers → stripers → stripes → tripes → tries → ties → tis → is → i
- manager → manger → mange → mane → man → an → a

To do:

- Describe in English (you can use pseudocode if you wish) how to determine if a word is a dropoff word using recursion.
- Write a program that implements your recursive algorithm. The algorithm should not use any global variables. Use the word list and binary search routine from program #2 to determine if a sequence of characters is a valid word in English. Hint: You will need some code that can make a copy of the current string with a letter removed at a given index.
- Loop through all 87337 words in the word list, testing each word with your recursive dropoff function, and find the longest word that is a dropoff word. If there is a tie you can pick one of the longest words.
- Modify the recursive algorithm so it prints out the dropoff words so you can determine the word sequence down to a single letter (the only one letter words are "i" and "a"). You can comment this out for other parts of the program if you don't want it to spam out a bunch of words. What is the dropoff sequence for the longest dropoff word?

5) (10 points) Taste of C++

Re-do Lab #5 except in C++ and you can use the `string` type. The output should be the same except in C++ instead.