Homework 1

GWU CSCI 1112 - Fall 2019

September 11, 2019

1 Introduction

This homework is intended to reinforce your understanding of working with one-dimensional arrays and to encourage you to think algorithmically. Your task is to develop an algorithm that solves the problem described using the provided framework.

1.1 Deadline

September 18, 2019 at 11:59pm

1.2 Submission

You must create a `.zip` file containing all of the files that you develop and work with and you must submit only the `.zip` file to blackboard before the deadline.

Your `.zip` file must be named using the following naming convention: `<yourNetId>-hw-01.zip`
For example, my NetId is `jrt`. If I submitted the homework, I would name my `.zip` file: `jrt-hw-01.zip`

1.3 Grading Rubric

- 60% For successful compilation
- 10% For correct approach
- 10% For correct output
- 10% For sufficient comments
- 10% For consistent coding style

1.4 Comments

I have provided relevant comments to document the file (a file header), the function signatures (function block comments), and inline comments. These comments are a model that you should follow in the future when developing files or functions on your own. This level of comments will be expected in future assignments.

You must substitute your name in the file header and provide any inline comments to document code that you contribute to your solution.

1.5 Plagarism

We will use a set of automated tools specifically designed to analyze code for plagarism. If you copy code from another source, classmate or website, there is a very high probability that these tools will flag your work as plagarized. You are permitted to discuss the problems at a high level; however, you must code your own solution. If you do not share code or outright borrow code from a website, you will have no problem with the plagarism filter.

2 Countdown

"Countdown" is an English gameshow. In one of the Countdown games, contestants are given a set of random letters which the contestants unscramble to find the longest word that can they can make using a subset from the randomly chosen letters. The more entertaining variant of this show is called "8 Out of 10 Cats Does Countdown" where comedians perform the same tasks.

For example, given the set of letters "ionsomsti", we can form the following words: "in", "insist", "mission", "mitosis", "omission". Out of these words, the longest word is "omission", so "omission" would be the winning word from the set of letters.

On "Countdown", the number of scrambled letters provided is 9 and therefore the longest words possible can contain 9 letters. For our solution, we will not bound the maximum number of characters in the set of letters. We will instead pass a string of letters and the string will dictate the maximum length of the available letters to choose from.

2.1 Instructions

This problem is similar to trying to find all anagrams in a dictionary for a given word or determining if a sentance is a pangram; however, it is slightly more challenging because there is no guarantee that it is possible to use all letters provided in the scrambled set of letters.

We will use the `WordTool` and dictionary of words to try to find the longest word possible to form from the scramble. I have edited the `words` file to remove entries from the dictionary that contain special characters like contractions and abbreviations, so you should use the `words` file provided in the assignment file.

A framework for this program has been provided in `Countdown.java`. You will implement three methods in the `Countdown` class:

- `WordGame`
- `contains`
- `Test3`

You will not alter any of the other code in the framework, you will use the provided interfaces for the above methods, and there is no need to add additional methods. There are three places that are marked with `TODO` that you will focus on.

Read the documentation associated with each method and fulfill the requirements outlined by that method's documentation. `WordGame` is intended to handle the selection of a word from the dictionary and then using the `contains` method to see if the word contains a subset of the letters provided. The `contains` method is intended to see if the word is contained in the set of letters.

The framework provides a set of "Unit Tests" to evaluate the algorithm. The first two unit tests are provided in the methods `Test1` and `Test2`. You will implement your own unit test in `Test3`. Your unit test must include a set of letters containing at least 9 unique letters and must contain at least 3 vowels. Your program must pass all three tests to be considered fully correct.

There may be a number of words with the same length that can be formed using the set of letters. We do not necessarily need to know all words that can be formed using the set of letters, so we will pick the first word in the alphabet that is the longest word. This offers a slight optimization but mostly it ensures that everyone gets the same answer to the first two unit tests.

2.2 Notes

A function `clean` has been provided that will remove any non-alpha characters from a string, will lowercase the cleaned string, and will return the lowercase, cleaned string. This method will ensure both that only legal letters are evaluated and will eliminate edge cases that we really don't want to deal with when preparing to compare string information.

The methods are documented using a style defined by Doxygen. This format allows documentation to be automatically generated from source code. Note how this format documents what a function does, what each of the inputs represents, and what the return value represents. In the future, your programs must include documentation of functions, and your function documentation must meet these three criteria. You will not be expected to conform to the Doxygen format, but you will not be discouraged from doing so.

2.3 Hint

The best approach to solve this problem was discussed in class as an approach to solving the `StringExample` problems in Module 1. While most of this document talks about strings, the best solution to this problem involves using arrays, so this problem is also about converting from one problem into another problem that is much easier to solve.