ECM1400 Programming

Date set: N/A

Hand-in date: N/A

This problem is designed to apply the concepts that we will cover in the first week of term, particularly flow charts and pseudo code, on a more substantial problem similar to what you will encounter in CA1. This specification also reflects the same structure that you can expect in CA1.

Specification

Countdown is a British TV game show broadcast by Channel 4. As part of the game, contestants are challenged to make the longest possible words by rearranging a set of letters selected by one of the contestants. In this mock coursework you are going to design an algorithm to simluate the game from a command-line interface (do not use a GUI in the design) and as pasrt of that algorithm produce a solver algoirthm that finds and reports the best answers.

`https://www.youtube.com/watch?v=k2owgezZRQM`

The simulation should be broken down into different sections and designed with flowcharts and pseudo code, strictly using the same identifiers for nested algorithms described below. Start by drawing out the flowcharts then translate the flowcharts to pseudo code after the flowcharts have been shared and the design agreed among your group.

- **Random character selector**

  Design a function (algorithm) called `select_characters` that asks the user to input a 'c' for a consonant or a 'v' for a vowel nine times and returns a string with the corresponding number of random consonant and vowel characters.

  [10 marks]

- **Dictionary reader**

  Design a function (algorithm) called `dictionary_reader`. The function should open the a file, `'words.txt'` is an example, iterate through each line to create a list of the words in the file that includes all the possible correct words for the game. `'words.txt'` can be downloaded from the ELE page so you can see the structure of the file.

[10 marks]

- **Word look up algorithm**

  Design a function (algorithm) called `word_lookup`. The function should check if the characters in a string, such as the string created by character selector, can be used to make any of the words in the `'words.txt'` file provided on ELE. Note characters cannot be resampled. The longest word, and any other words of that length, found in the `'words.txt'` file that can be made from the characters in the string argument should be returned in a list.

[10 marks]

- **Program design**

  Write a parent algorithm, that includes and utilises the functions (nested algorithms) described above to simulate the letters game from the Countdown game show. The algorithm should ask the user nine times to select a consonant or vowel characters before reporting a set of nine corresponding randomly generated letters. The algorithm should then then give the user a chance to input their best guess at the longest word that can be created from those letters. The users guess should then be checked and the number of points they scored should be reported back to the user. The number of points is the length of a correct answer. Finally the 'best' answers, the longest words that can be made from the characters, should be computed by the program and printed, that is all the longest possible words that can be made from the characters selected.

  As a stretch goal for the assignment there are a set of bonus features to improve the game simulation. These features will require additional research beyond the content of the learning materials to implement:

  - **Accurate probability distribution:** The Countdown game show has an uneven number of each letter on their piles of consonants and vowels. Make your random function reflect the non-standard probability distribution that is used in the real game (google can be used to find the letter representation).
  - **User input validation:** Users can be unpredictable. If the user inputs something unexpected you should ask for another input until they enter one of the options correctly.
  - **ASCII art user interface:** Command-line interfaces focus on functionality by using basic formatting which often isn't the most visually appealing. However, using ascii art can be a way to make them slightly more fun at the beginning and the end of a program without interfering with the functionality. For algorithm design this isn't necessary but will come up when we implement this in Python.
  - **A countdown:** The countdown game show on TV is centered around a physical 'countdown' which lasts 30 seconds. This is usually done using threads which are beyond the scope of this course but it is feasible that this could be implemented with some online research.
  - **A testing framework:** Any good software has a full testing phase before code is deployed and executed. You don't need to do a full testing suite but provide a single test case for each function and print the outcomes of the tests (pass or fail) each time the program is run.

[20 marks]

You should carefully follow the structure and use the identifier names described above. If you design a program with a different structure you will not attract as many marks (this is a mock

assignment and won't be marked but expect to see this in a real assignment).

This is only a one player simulation and only needs to have one round.