

# ECM1400 Programming Continuous Assessment 1

Date set: 30th September, 2019

Hand-in date: **11:59am Friday 18th October, 2019**

This continuous assessment (CA) comprises 15% of the overall module assessment.

Note that both paper (BART) and electronic submissions are required and instructions are provided at the end of the specification. The specification will be discussed in the lecture on Tuesday 1st October 2019. Students have 7 days (until 7th October) to raise any questions regarding the specification either in lectures or by email. Up to this date the module leader may make adaptations to clarify the assessment. After this date the specification will not change.

---

This CA is designed to test the programming concepts that we will cover in the first three weeks of term, particularly variables, functions, loops, lists and conditionals.

## 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 coursework you are going to simulate the game from a command-line interface (do not use a GUI) and produce a solver that finds and reports the best answers.



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

The simulation should be broken down into different sections and implemented using the programming constructs, strictly using the same identifiers, described below. Marks are available for each section as well as the overall functionality of your program:

- **Random character selector**

Write a function 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**

Write a function `dictionary_reader` that takes a string argument containing a filename. The function should open the relevant file, iterate through each line to create a list of the words in the file and return the list. Use this function to read a file called `'words.txt'` which can be downloaded from the coursework ELE page.

[10 marks]

- **Word look up algorithm**

Write a function called `word_lookup` that takes a string argument. The function should check if the characters in the string are contained in any of the words in the `'words.txt'` file provided on ELE. All words that exist in the `'words.txt'` file containing the characters from the string argument should be returned in a list.

[10 marks]

*Note: An algorithm for this section will be discussed in the lectures during teaching week 2. In summary, each word in should be sorted alphabetically before being compared to the sorted string argument.*

- **Program design**

Write a program, in a file called `countdown.py`, that includes and utilises the functions described above to simulate the letters game from the Countdown game show. The program should ask the user nine times to select a consonant or vowel characters before reporting the a set of nine corresponding randomly generated letters and then giving 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 should also computed by the program be 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.
- **A countdown:** The countdown game show on TV is centered around a physical 'countdown' which lasts 30 seconds. It isn't possible to implement this in a robust way without 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 only a one player simulation. The expected behaviour of the game will be provided on ELE and demonstrated in the lectures in week 2.

## Submitting your work

The CA requires both paper and electronic submissions.

**Paper** You should submit paper copies of the code and any output for **all** the questions to the Harrison Student Services Office in the foyer of the Harrison Building by the deadline of **11:59am Friday 18th October, 2019**. Markers will not be able to give feedback if you do not submit hardcopies of your code and marks will be deducted if you fail to do so.

Paper submissions should have the BART cover sheet securely attached to the front and should be anonymous (that is, the marker should not be able to tell who you are from the submission). If this is the first time you have used BART, please make sure that you understand the procedure beforehand and leave plenty of time as there are often queues close to the deadline.

Where you are asked for paper copies of the output of your code, please cut and paste the output from the terminal rather than taking a screenshot, because the screenshot is often illegible after printing.

**Electronic** You should submit the files containing the code for each question via the electronic submission system at <http://empslocal.ex.ac.uk/submit/>. You should use the category containing ECM1400 and CA1. If you want to submit multiple files, I don't expect multiple files for CA1, use **zip** to compress your files into a single file, and upload this file using the submit system. You must do this by the deadline.

You will be sent an email by the submit system asking you to confirm your submission by following a link. Your submission is not confirmed until you do this. It is best to do it straightaway, but there is a few hours leeway after the deadline has passed. It is possible to unsubmit and resubmit electronic coursework — follow the instructions on the submission website.