

Programming Assignment 2

*University of Wisconsin – La Crosse**Due Date: March 28*

Omnibetum

Description: For this assignment, you will implement a single-player word game. The idea is simple: the player is presented with 7 distinct letters from the English alphabet. The goal is to make as many words as possible using the 7 letters. You will implement a console version of this game. That's it – couldn't be simpler.

Details: Well, maybe it could be a little simpler. As always the devil is in the details, as the list below illustrates.

- One of the 7 letters is distinguished and must appear in every word. Any word the user enters without the distinguished letter is considered invalid and does not score.
- Only words of length at least 4 are acceptable.
- Letters may be repeated within a word. Each instance of a letter counts toward the length of the word. This includes the distinguished letter.

A word that uses all 7 letters at least once is called an **omnibetum**. Scoring is as follows:

| Number of letters (n) | Points |
|---------------------------|---------|
| $n < 4$ | 0 |
| $n = 4$ | 1 |
| $n > 4$ | n |
| omnibetum | $n + 7$ |

Game play: Playing the game is straightforward. Your program will print instructions for the user (see details below) and present a menu that includes a prompt for them to enter a word. It will then evaluate the word to determine the number of points awarded and inform the player of that number. If the word is not accepted, your program will inform the user of that as well. Entry of the words can be in any combination of upper and lower case. Play continues until the user indicates that they wish to end the game. When this occurs, the program should list all of the valid words entered by the user

and their score. All output generated by the program should be well-formatted and easily read.

In addition, the user can request their score; the list of words they have correctly guessed, presented in alphabetical order; to shuffle the letters, which rearranges the order in which they are displayed for the user; or help, in the form of the instructions that were presented at the beginning of the game . For these menu options, you MUST use '1', '2', '3', and 'h', respectively. To quit the game, you MUST use 'q'.

When displaying the characters used to form words, the distinguished character should always appear last and should be marked with asterisks (for example, *B*).

Coding details: Functional decomposition is an important aspect of this assignment. Give some thought to how to decompose the code into functions such that each accomplishes an identifiable task. Also think carefully about how to represent the data your program must maintain.

Look carefully at the example run at the end of this document to see how the game progresses and how users interact with it.

To make game output easier to read, I use a function that clears the screen. I have not been able to verify this function on Windows. The code for the function is:

```
def clear_console():
    command = 'clear'
    if os.name in ['nt', 'dos']:
        command = 'cls'

    os.system(command)
```

Input: Your program will read data from one of several input files. Each file contains the 7 letters to be used to create words, the number of words that are considered valid, and the valid words. The distinguished letter is the last. Note that the list of valid words is not exhaustive. In particular, words that are particularly obscure, are proper nouns, or may be considered offensive are not included.

When run, your program should present a list of the input files to the user and allow them to choose which to use. This choice should be indicated by entry of an integer value with 1 being the first file, etc. Ensure that an invalid entry does not crash the program and that the user is asked to try again until they enter a correct value.

When the user enters a word, your program should check that it is valid (i.e., in the list of words you read from the input file. If it is, the program will first check to see if the user has guessed that word previously and print a message informing the user if that's the case. Otherwise, it will calculate the score for the word, inform the user,

and update data values as appropriate. If the word is not valid, the program should print a message indicating that.

Your program should be robust against invalid menu selections. If a user enters incorrect input, the program should inform them and continue running with their next choice.

Information on submitting your program appears below.

Adhere to common coding conventions and **comment your code**.. Include a comment at the top that looks like this:

```
#
# CS 224 Spring 2022
# Programming Assignment 2
#
# Description of what the program does
#
# Author: Your name here
# Date: March XX, 2022
#
```

Submission: The name of your program must be `omnibetum.py`. The program must be in a directory named `Lastname-Prog02` where `Lastname` is replaced with your last name. DO NOT include the datafiles in your submission. You must zip your directory prior to submission. Submit your solution by 11:59 PM on the due date.

Sample output:

Welcome to Omnibetum - the CS224 word game.

Omnibetum is a single-player word game. You will be presented with seven letters from which you will form as many words as possible. One of the letters is marked with asterisks. It must appear in every word you enter. You may repeat letters within a word. Words with fewer than 4 letters will receive 0. Scoring for other words is as follows

4 letters: 1 point

n letters: n points (for $n > 4$)

A word that uses all seven letters is an Omnibetum and scores a bonus of 7 points.

Good luck!

Please enter your name: Deadpool

You will now choose the game to play. Your selection must be the integer part of the inputs listed below.

input1.txt input2.txt input3.txt input4.txt input5.txt
input6.txt input7.txt

Enter the game to play: 1

A L E B I P *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: alibi

That word is not valid.

A L E B I P *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: clip

Good. You scored 1 point.

A L E B I P *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: 3

Shuffling letters...

A E L B P I *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: cable

Great word. You scored 5 points.

A E L B P I *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: applicable

That's an omnibetum! You scored 17 points.

A E L B P I *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: 2

Your words are:

applicable cable clip

A E L B P I *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: 1

Deadpool, your score is: 23

A E L B P I *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: callable

Great word. You scored 8 points.

A E L B P I *C*

Player options:

- 1: Print your score
- 2: Print your valid words
- 3: Shuffle the letters
- h: Help
- q: Quit game

Or enter a word

Please enter a word or menu option: q

You found 4 valid words for a score of 31.

Your words are:

applicable cable callable clip

Thank you for playing, Deadpool. Goodbye.