3. WRITTEN RESPONSES

3 a.

3.a.i.

The program’s purpose is to increase the user’s creative-thinking and problem-solving skills in an interpretive-manner through a game similar to Wordle©□.

3.a.ii.

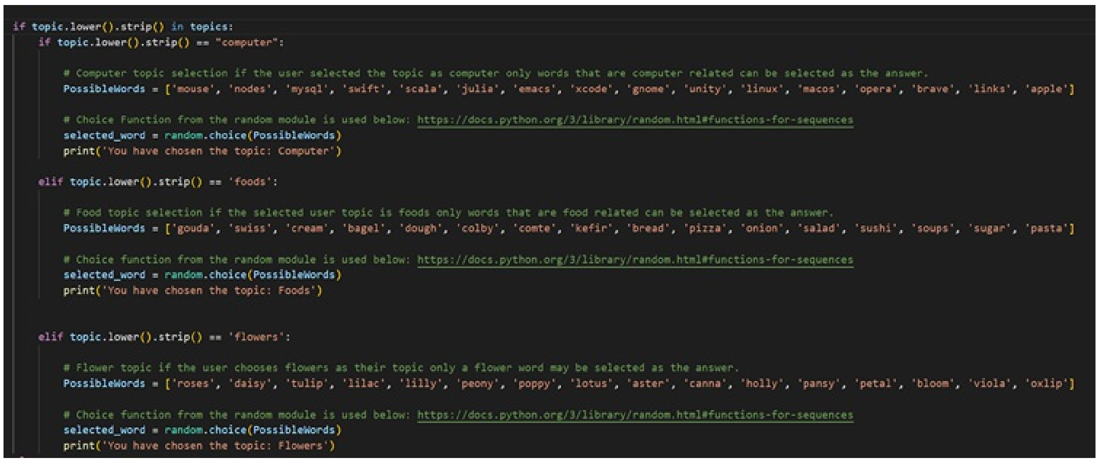
The program’s function is that a user is able to enter a 5-character-long-word in their corresponding word-bank, which is displayed in an order corresponding to their turn in a grid, if a letter in the input-word matches the position and letter in the randomly-chosen-word it will be bolded and if it's only present it will be italicized to inform the user, this will happen every turn. If the user wins by guessing before their 5-turns are up a winning-message is displayed and if not a losing-message.

3.a.iii.

The inputs are the 5-letter words the user inputs such as “gouda”, “cream”, etc. and are in the corresponding word-bank otherwise the program will output an error normally the wordle-grid after each turn is updated with the user's guess and the appropriate-character in the word is bolded if it's in the correct-location and matches or italicized if only in the chosen-word. At the end of the user’s guesses, the user is told they lost, and if the user guesses correctly they are informed they won.

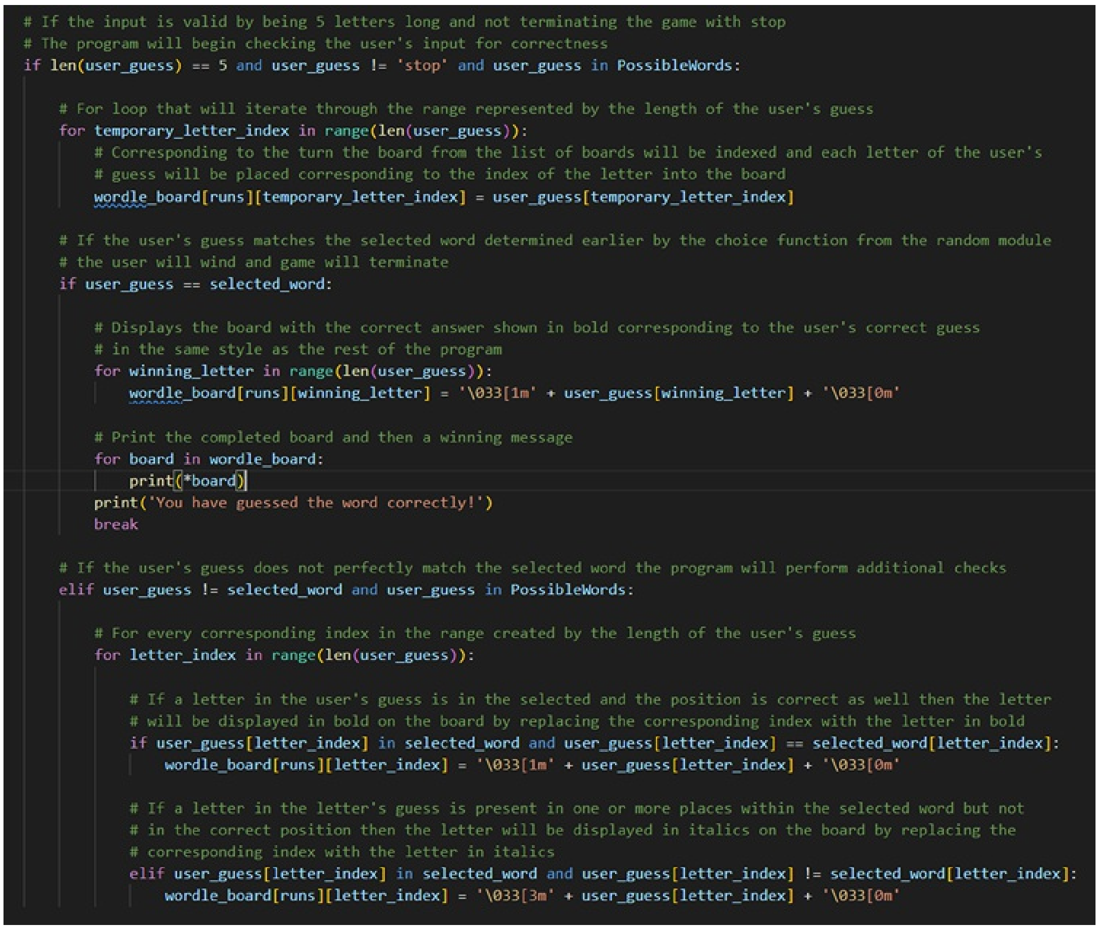
3 b.

3.b.i.



```python
if topic.lower().strip() in topics:
    if topic.lower().strip() == "computer":

        # Computer topic selection if the user selected the topic as computer only words that are computer related can be selected as the answer.
        PossibleWords = ['mouse', 'nodes', 'mysql', 'swift', 'scala', 'julia', 'emacs', 'xcode', 'gnome', 'unity', 'linux', 'macos', 'opera', 'brave', 'links', 'apple']

        # Choice Function from the random module is used below: https://docs.python.org/3/library/random.html#functions-for-sequences
        selected_word = random.choice(PossibleWords)
        print('You have chosen the topic: Computer')

    elif topic.lower().strip() == 'foods':

        # Food topic selection if the selected user topic is foods only words that are food related can be selected as the answer.
        PossibleWords = ['gouda', 'swiss', 'cream', 'bagel', 'dough', 'colby', 'comte', 'kefir', 'bread', 'pizza', 'onion', 'salad', 'sushi', 'soups', 'sugar', 'pasta']

        # Choice function from the random module is used below: https://docs.python.org/3/library/random.html#functions-for-sequences
        selected_word = random.choice(PossibleWords)
        print('You have chosen the topic: Foods')


    elif topic.lower().strip() == 'flowers':

        # Flower topic if the user chooses flowers as their topic only a flower word may be selected as the answer.
        PossibleWords = ['roses', 'daisy', 'tulip', 'lilac', 'lilly', 'peony', 'poppy', 'lotus', 'aster', 'canna', 'holly', 'pansy', 'petal', 'bloom', 'viola', 'oxlip']

        # Choice function from the random module is used below: https://docs.python.org/3/library/random.html#functions-for-sequences
        selected_word = random.choice(PossibleWords)
        print('You have chosen the topic: Flowers')
```

3.b.ii.



```python
# If the input is valid by being 5 letters long and not terminating the game with stop
# The program will begin checking the user's input for correctness
if len(user_guess) == 5 and user_guess != 'stop' and user_guess in PossibleWords:

    # For loop that will iterate through the range represented by the length of the user's guess
    for temporary_letter_index in range(len(user_guess)):
        # Corresponding to the turn the board from the list of boards will be indexed and each letter of the user's
        # guess will be placed corresponding to the index of the letter into the board
        wordle_board[runs][temporary_letter_index] = user_guess[temporary_letter_index]

    # If the user's guess matches the selected word determined earlier by the choice function from the random module
    # the user will wind and game will terminate
    if user_guess == selected_word:

        # Displays the board with the correct answer shown in bold corresponding to the user's correct guess
        # in the same style as the rest of the program
        for winning_letter in range(len(user_guess)):
            wordle_board[runs][winning_letter] = '\033[1m' + user_guess[winning_letter] + '\033[0m'

        # Print the completed board and then a winning message
        for board in wordle_board:
            print(*board)
        print('You have guessed the word correctly!')
        break

    # If the user's guess does not perfectly match the selected word the program will perform additional checks
    elif user_guess != selected_word and user_guess in PossibleWords:

        # For every corresponding index in the range created by the length of the user's guess
        for letter_index in range(len(user_guess)):

            # If a letter in the user's guess is in the selected and the position is correct as well then the letter
            # will be displayed in bold on the board by replacing the corresponding index with the letter in bold
            if user_guess[letter_index] in selected_word and user_guess[letter_index] == selected_word[letter_index]:
                wordle_board[runs][letter_index] = '\033[1m' + user_guess[letter_index] + '\033[0m'

            # If a letter in the letter's guess is present in one or more places within the selected word but not
            # in the correct position then the letter will be displayed in italics on the board by replacing the
            # corresponding index with the letter in italics
            elif user_guess[letter_index] in selected_word and user_guess[letter_index] != selected_word[letter_index]:
                wordle_board[runs][letter_index] = '\033[3m' + user_guess[letter_index] + '\033[0m'
```

3.b.iii.

The name of the list used is “PossibleWords” the contents of which vary depending on which of the 3-modes is selected.

3.b.iv.

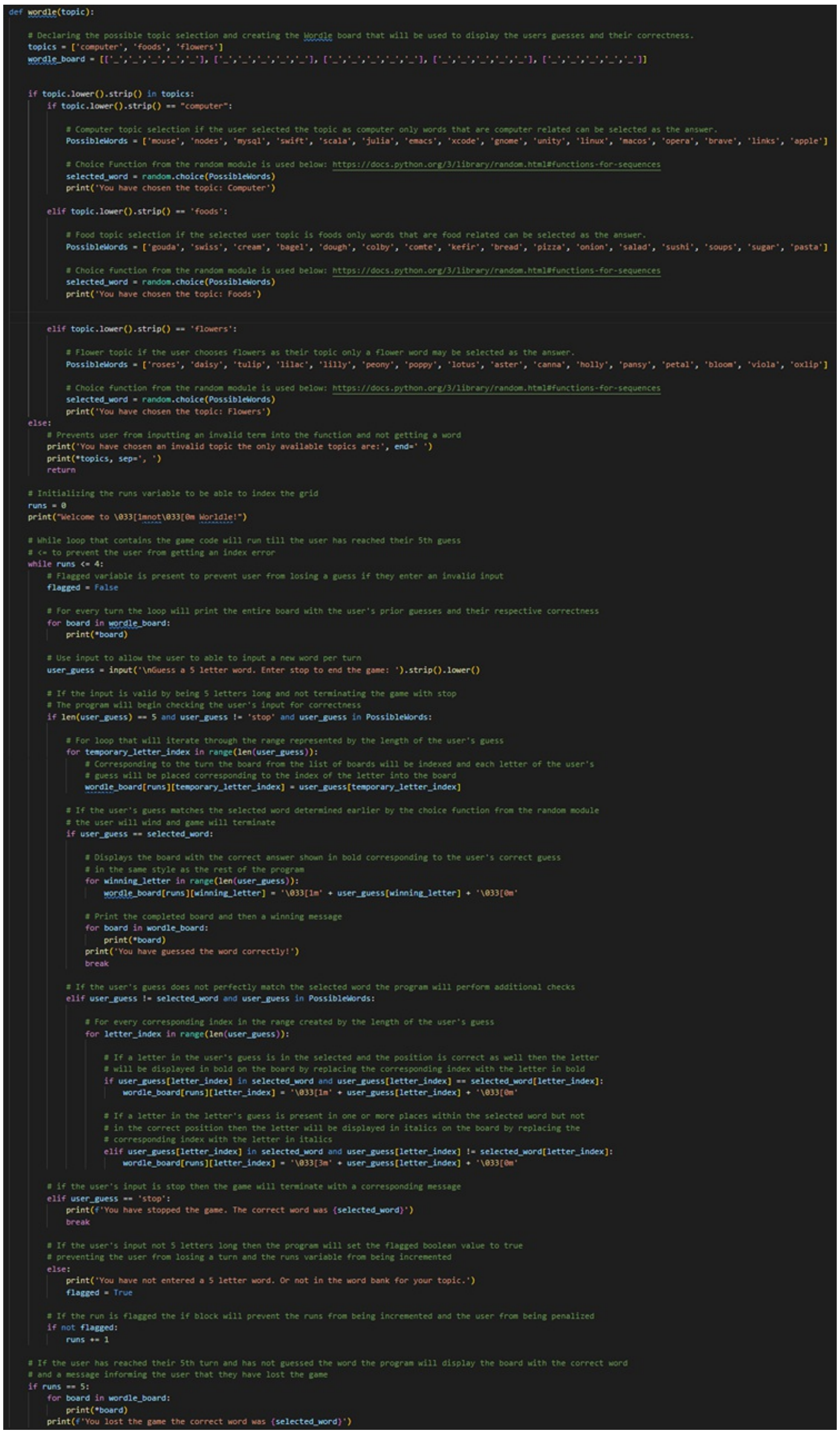
The data in the list “PossibleWords” across all three-versions is a collection of 16-strings that represent all the possible words that could be chosen as the answer to the Wordle-game for any given topic.

3.b.v.

If “PossibleWords” wasn't used a variable needs to be assigned to every one of the 16-strings per-category, making category-selection an inefficient and time-consuming process. With input-validation from the user to screen if they inputted a word within their word-bank being almost impossible with every variable having to be individually compared to the input and additional verification to only compare the variables from their topic. Also, the lack of a list would also prevent the main-functionality of the program by making selecting a random-word practically impossible. Which is far more inefficient than where a list stores all the various strings, can easily replace them all for per-topic, an efficient way to see if the input is in their word-list preventing invalid-inputs, and allowing for an item to be easily selected as the winning-word allowing for the game to work.

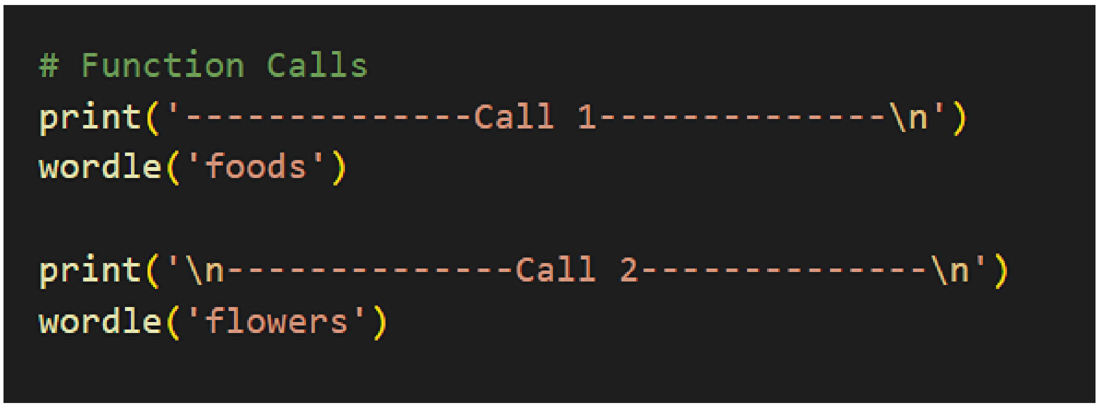
3 c.

3.c.i.



```python
def wordle(topic):

    # Declaring the possible topic selection and creating the Wordle board that will be used to display the users guesses and their correctness.
    topics = ['computer', 'foods', 'flowers']
    wordle_board = [['_','_','_','_','_'], ['_','_','_','_','_'], ['_','_','_','_','_'], ['_','_','_','_','_'], ['_','_','_','_','_']]


    if topic.lower().strip() in topics:
        if topic.lower().strip() == "computer":

            # Computer topic selection if the user selected the topic as computer only words that are computer related can be selected as the answer.
            PossibleWords = ['mouse', 'nodes', 'mysql', 'swift', 'scala', 'julia', 'emacs', 'xcode', 'gnome', 'unity', 'linux', 'macos', 'opera', 'brave', 'links', 'apple']

            # Choice Function from the random module is used below: https://docs.python.org/3/library/random.html#functions-for-sequences
            selected_word = random.choice(PossibleWords)
            print('You have chosen the topic: Computer')

        elif topic.lower().strip() == 'foods':

            # Food topic selection if the selected user topic is foods only words that are food related can be selected as the answer.
            PossibleWords = ['gouda', 'swiss', 'cream', 'bagel', 'dough', 'colby', 'comte', 'kefir', 'bread', 'pizza', 'onion', 'salad', 'sushi', 'soups', 'sugar', 'pasta']

            # Choice function from the random module is used below: https://docs.python.org/3/library/random.html#functions-for-sequences
            selected_word = random.choice(PossibleWords)
            print('You have chosen the topic: Foods')


        elif topic.lower().strip() == 'flowers':

            # Flower topic if the user chooses flowers as their topic only a flower word may be selected as the answer.
            PossibleWords = ['roses', 'daisy', 'tulip', 'lilac', 'lilly', 'peony', 'poppy', 'lotus', 'aster', 'canna', 'holly', 'pansy', 'petal', 'bloom', 'viola', 'oxlip']

            # Choice function from the random module is used below: https://docs.python.org/3/library/random.html#functions-for-sequences
            selected_word = random.choice(PossibleWords)
            print('You have chosen the topic: Flowers')
    else:
        # Prevents user from inputting an invalid term into the function and not getting a word
        print('You have chosen an invalid topic the only available topics are:', end=' ')
        print(*topics, sep=', ')
        return

    # Initializing the runs variable to be able to index the grid
    runs = 0
    print("Welcome to \033[1mnot\033[0m Worldle!")

    # While loop that contains the game code will run till the user has reached their 5th guess
    # <= to prevent the user from getting an index error
    while runs <= 4:
        # Flagged variable is present to prevent user from losing a guess if they enter an invalid input
        flagged = False

        # For every turn the loop will print the entire board with the user's prior guesses and their respective correctness
        for board in wordle_board:
            print(*board)

        # Use input to allow the user to able to input a new word per turn
        user_guess = input('\nGuess a 5 letter word. Enter stop to end the game: ').strip().lower()

        # If the input is valid by being 5 letters long and not terminating the game with stop
        # The program will begin checking the user's input for correctness
        if len(user_guess) == 5 and user_guess != 'stop' and user_guess in PossibleWords:

            # For loop that will iterate through the range represented by the length of the user's guess
            for temporary_letter_index in range(len(user_guess)):
                # Corresponding to the turn the board from the list of boards will be indexed and each letter of the user's
                # guess will be placed corresponding to the index of the letter into the board
                wordle_board[runs][temporary_letter_index] = user_guess[temporary_letter_index]

            # If the user's guess matches the selected word determined earlier by the choice function from the random module
            # the user will wind and game will terminate
            if user_guess == selected_word:

                # Displays the board with the correct answer shown in bold corresponding to the user's correct guess
                # in the same style as the rest of the program
                for winning_letter in range(len(user_guess)):
                    wordle_board[runs][winning_letter] = '\033[1m' + user_guess[winning_letter] + '\033[0m'

                # Print the completed board and then a winning message
                for board in wordle_board:
                    print(*board)
                print('You have guessed the word correctly!')
                break

            # If the user's guess does not perfectly match the selected word the program will perform additional checks
            elif user_guess != selected_word and user_guess in PossibleWords:

                # For every corresponding index in the range created by the length of the user's guess
                for letter_index in range(len(user_guess)):

                    # If a letter in the user's guess is in the selected and the position is correct as well then the letter
                    # will be displayed in bold on the board by replacing the corresponding index with the letter in bold
                    if user_guess[letter_index] in selected_word and user_guess[letter_index] == selected_word[letter_index]:
                        wordle_board[runs][letter_index] = '\033[1m' + user_guess[letter_index] + '\033[0m'

                    # If a letter in the letter's guess is present in one or more places within the selected word but not
                    # in the correct position then the letter will be displayed in italics on the board by replacing the
                    # corresponding index with the letter in italics
                    elif user_guess[letter_index] in selected_word and user_guess[letter_index] != selected_word[letter_index]:
                        wordle_board[runs][letter_index] = '\033[3m' + user_guess[letter_index] + '\033[0m'

        # if the user's input is stop then the game will terminate with a corresponding message
        elif user_guess == 'stop':
            print(f'You have stopped the game. The correct word was {selected_word}')
            break

        # If the user's input not 5 letters long then the program will set the flagged boolean value to true
        # preventing the user from losing a turn and the runs variable from being incremented
        else:
            print('You have not entered a 5 letter word. Or not in the word bank for your topic.')
            flagged = True

        # If the run is flagged the if block will prevent the runs from being incremented and the user from being penalized
        if not flagged:
            runs += 1

    # If the user has reached their 5th turn and has not guessed the word the program will display the board with the correct word
    # and a message informing the user that they have lost the game
    if runs == 5:
        for board in wordle_board:
            print(*board)
        print(f'You lost the game the correct word was {selected_word}')
```

3.c.ii.



```
# Function Calls
print('-------------Call 1-------------\n')
wordle('foods')

print('\n-------------Call 2-------------\n')
wordle('flowers')
```

3.c.iii.

“wordle” has a parameter-string used for topic selection based on which a corresponding 16-word-list is chosen. A 2d-list representing the board is also present for every non-flagged-run of the while-loop will be indexed and change all the characters to ones from the user's input on the level that corresponds to the run, check if the user guessed correctly, or has a letter in the exact-position which will bold the letter in the 2d-list or just present in the original word which will italicize it. Once the run is over the 2d-list will be printed, if the user runs out of guesses(<=4) or type “stop” the program will end. Allowing the user to play the game whenever they please without having to understand complex-program-code.

3.c.iv.

If the parameter-topic is in the topic-list a corresponding selection-statement will execute, setting the word-list and selecting a random word from it and setting runs equal to 0. Afterwards, a while-loop containing the game-code is run per-run and a 2d-list containing the user’s prior guesses will be printed using a for-loop within the while-loop and a Flagged variable will equal false. If the selection-statement determines that the length of the input-word is greater-than-5, not “stop”, and in the word-list the characters from the input-word replace the corresponding indexes in the level of the 2d-list-”wordle_board” through a for-loop with level based on their “runs”-variable and indexes from a range with a length of the input-word (runs-variable is incremented by one per-loop), then if the selection-statement determines users-guess matches the chosen-word letters of the word in the grid are bolded, the loop ends and they win, otherwise a different selection-statement will be executed and compare each index of the input-word and the selected-word and if the index and letter match it's bolded in the 2d-list and if the letter is only in the word another selection-statement will italicize it if neither letter will be left unchanged. If input is invalid they will be warned Flagged will equal true, the runs variable not be incremented and the while-loop will continue. Also if the user inputs “stop” the program terminates.

3 d.

3.d.i.

First call:

wordle('foods')

Second call:

wordle('flowers')

3 d.ii.

Condition(s) tested by first call:

It tests if the parameter “foods” will correctly define the list-”PossibleWord” with food-words and randomly select the “selected_word” from it, and validates the user-inputs using them.

Condition(s) tested by second call:

It tests if the parameter “flowers” will correctly define the list-”PossibleWord” with flower-words and randomly select the “selected_word” from it, and validates the user-inputs using them.

3.d.iii.

Results of the first call:

A welcome-message and that the foods-topic has been selected will be displayed. Then the grid will print, afterwards an input is prompted after which output vary but inputs must be food-words.

Results of the second call:

A welcome-message and that the flowers-topic has been selected will be displayed. Then the grid will print, afterwards an input is prompted after which outputs vary but inputs must be flower-words.