Pseudocode for admin.py

Name: Julian Mendis

Student Number: 10685665

================================================================================

MAIN PROGRAM

================================================================================

Try to open data.txt in read mode

Load JSON data from file into data variable

Close file

If any exceptions occur

Set data to empty list

Display line of equals signs

Display "WELCOME TO ONE MUST GO - ADMIN PROGRAM"

Display "Created by Julian Mendis"

Display line of equals signs

Loop endlessly

Display blank line

Display menu prompt: "Choose [a]dd, [l]ist, [s]earch, [v]iew, [d]elete or [q]uit."

Prompt user to enter their choice

Convert choice to lowercase and strip whitespace

If choice is a

Display "ADD NEW CATEGORY" heading

Call input\_something function with "Enter category name: " prompt

Store result in category\_name variable

Create list of existing category names in lowercase from data list

If category\_name in lowercase exists in existing names list

Display error message about duplicate category name

Continue to next iteration of main loop

Create empty options list

Loop while length of options list is less than 5

If length of options list is less than 2

Call input\_something function with prompt showing option number

Store result in option\_name variable

Otherwise

Prompt user for option name or Enter to finish

Strip whitespace from input

If input is empty and options list has at least 2 items

Break out of options loop

If input is empty and options list has fewer than 2 items

Display error about needing at least 2 options

Continue to next iteration of options loop

Set option\_name to the input value

Create list of existing option names in lowercase from options list

If option\_name in lowercase exists in existing options list

Display error message about duplicate option

Continue to next iteration of options loop

Append option\_name to options list

If length of options list equals 5

Display "Maximum of 5 options reached" message

Break out of options loop

Create new\_category dictionary with two keys

Set name key to category\_name value

Set options key to list comprehension creating dictionaries for each option

Each option dictionary has name key set to option and votes key set to 0

Append new\_category dictionary to data list

Call save\_data function passing data as parameter

Display success message with category name

Otherwise if choice is l

Display "ALL CATEGORIES" heading

If data list is empty

Display "No categories saved" message

Otherwise

Loop through data list using enumerate to get index and category

Get length of category options list and store in option\_count

Display index plus 1, category name, and option count in parentheses

Otherwise if choice is s

Display "SEARCH CATEGORIES" heading

If data list is empty

Display "No categories saved" message

Otherwise

Call input\_something function with "Enter search term: " prompt

Convert result to lowercase and store in search\_term

Set results\_found flag to False

Loop through data list using enumerate to get index and category

If search\_term is contained in category name converted to lowercase

Get length of category options list and store in option\_count

Display index plus 1, category name, and option count in parentheses

Set results\_found flag to True

If results\_found flag is False

Display "No results found" message

Otherwise if choice is v

Display "VIEW CATEGORY" heading

If data list is empty

Display "No categories saved" message

Otherwise

Call input\_int function with prompt and data list length as max value

Store result in index\_num variable

Get category from data list at index of index\_num minus 1

Display blank line

Display "Category: " followed by category name

Display "Options" with count in parentheses

Loop through each option in category options list

Display option name and vote count with proper formatting

Otherwise if choice is d

Display "DELETE CATEGORY" heading

If data list is empty

Display "No categories saved" message

Otherwise

Call input\_int function with delete prompt and data list length as max value

Store result in index\_num variable

Get category name from data list at index of index\_num minus 1

Prompt user to confirm deletion with y/n

Convert confirmation to lowercase

If confirmation equals y

Delete item from data list at index of index\_num minus 1

Call save\_data function passing data as parameter

Display "Category deleted" message

Otherwise

Display "Deletion cancelled" message

Otherwise if choice is q

Display blank line

Display thank you message including program name and author name

Display "Goodbye!" message with blank line

Break out of main loop to end program

Otherwise

Display "Invalid choice. Please try again." message

Program ends

================================================================================

FUNCTION DEFINITIONS

================================================================================

Function: input\_something

Parameters: prompt (the message to display when asking for input)

Returns: String containing validated user input with whitespace removed

Loop endlessly

Prompt user for input using prompt parameter

Strip whitespace from beginning and end of input

Store result in user\_input variable

If user\_input is not empty

Return user\_input string

End function

--------------------------------------------------------------------------------

Function: input\_int

Parameters: prompt (the message to display), max\_value (maximum acceptable value)

Returns: Integer between 1 and max\_value inclusive

Loop endlessly

Try to execute the following

Prompt user for input using prompt parameter

Convert input to integer and store in value variable

If value is between 1 and max\_value inclusive

Return value as integer

Otherwise

Display message asking for number between 1 and max\_value

If ValueError exception occurs

Display "Invalid input. Please enter a valid number." message

End function

--------------------------------------------------------------------------------

Function: save\_data

Parameters: data (the list of category dictionaries to save to file)

Returns: Nothing

Open data.txt file in write mode

Write data to file in JSON format with 4-space indentation

Close file

End function

```

---

## Pseudocode for omg.py (NOT REQUIRED - For Reference Only)

```

Pseudocode for omg.py (GUI Program)

Name: Julian Mendis

Student Number: 10685665

Note: Pseudocode is not required for GUI programs according to assignment brief.

This is provided for reference only and should NOT be submitted.

================================================================================

CLASS: ProgramGUI

================================================================================

Constructor Method: \_\_init\_\_

Parameters: self

Returns: Nothing

Try to execute the following

Open data.txt file in read mode

Load JSON data from file into self.data attribute

Close file

If any exceptions occur

Display error messagebox with title "Error"

Display message about missing or invalid file

Return from constructor to end program

If self.data list is empty

Display error messagebox with title "Error"

Display message about no categories found

Return from constructor to end program

Create main window and store in self.root attribute

Set window title to "One Must Go"

Set window geometry to 600x500 pixels

Set window background color to dark blue

Set self.index attribute to 0

Create header\_frame with dark background and fixed height

Pack header\_frame at top with padding

Disable frame propagation

Create title\_label in header\_frame

Set text to "ONE MUST GO" in large bold white font

Pack title\_label to expand in frame

Create subtitle\_label in header\_frame

Set text to question about living without options

Pack subtitle\_label below title

Create self.category\_label for displaying category names

Set font to large bold blue text

Set text wrapping width

Pack label with vertical padding

Create self.options\_frame for holding option buttons

Set background to dark color

Pack frame to fill and expand with padding

Create footer\_frame with dark background and fixed height

Pack footer\_frame at bottom

Disable frame propagation

Create footer\_label in footer\_frame

Set text to "Click on your choice"

Pack label to expand in frame

Call show\_category method

Call mainloop method on self.root to start GUI

End constructor

--------------------------------------------------------------------------------

Method: show\_category

Parameters: self

Returns: Nothing

Get current category from self.data at index self.index

Store in current\_category variable

Update self.category\_label text to show current category name

Loop through all widgets in self.options\_frame

Destroy each widget to clear previous buttons

Loop through each option in current category options list

Get option name from option dictionary

Create button in self.options\_frame

Set button text to option name

Set font to medium size

Set colors for normal and active states

Set padding and cursor style

Set command to call record\_vote method with option name using lambda

Pack button to fill horizontally with vertical padding

Bind mouse enter event to change button background color

Bind mouse leave event to restore button background color

End method

--------------------------------------------------------------------------------

Method: record\_vote

Parameters: self, name (the name of the selected option)

Returns: Nothing

Get current category from self.data at index self.index

Loop through each option in current category options list

If option name matches name parameter

Add 1 to option votes value

Break out of loop

Open data.txt file in write mode

Write self.data to file in JSON format with indentation

Close file

Display info messagebox with title "Vote Recorded"

Display message confirming vote for option name

If self.index equals length of self.data minus 1

Display info messagebox with title "Complete"

Display thank you message with category count

Destroy self.root window to end program

Otherwise

Add 1 to self.index

Call show\_category method to display next category

End method

================================================================================

MAIN EXECUTION

================================================================================

If this file is run directly

Create instance of ProgramGUI class