Author: Shenali Fernando

Student ID: 21200717

Pseudo code for Task D and E

```
CLASS HistogramApp
  FUNCTION __init__(traffic_data, date)
    SET self.traffic_data TO traffic_data
    SET self.date TO date
    CREATE a new tkinter window (self.root)
    SET window title to "Traffic Histogram for {date}"
    SET self.canvas TO None

  FUNCTION setup_window()
    CREATE a canvas (self.canvas) with width 1200, height 600, and background color "#FEF5E7"
    PACK the canvas to fill the window
    CREATE text on canvas for title "Histogram of Vehicles Frequency per Hour ({self.date})"

  FUNCTION draw_histogram()
    DEFINE colors for junctions
    SET max_vehicle_count to the maximum count from traffic_data
    SET max_height to 300
    SET bar_width to 18
    SET group_width to the total width of bars for each hour
    SET x to 50 and y to 500

    DRAW x-axis line on canvas
    CREATE x-axis label "Hours 00:00 to 24:00"

    FOR each hour from 0 to 23
```

```
        SET x_start to the starting x position for the hour
        FOR each junction and its color
            GET vehicle_count for the current hour
            CALCULATE height of the bar based on vehicle_count
            DRAW rectangle (bar) on canvas for the junction
            CREATE text label on top of the bar showing vehicle_count

    FOR each hour from 0 to 23
        CREATE text label for the hour below the x-axis

FUNCTION add_legend()
    SET x_legend to 55 and y_legend to 70
    DEFINE legend items for junctions
    FOR each legend item
        DRAW rectangle for the legend item
        CREATE text label for the legend item

FUNCTION run()
    CALL setup_window()
    CALL draw_histogram()
    CALL add_legend()
    START tkinter main loop

CLASS MultiCSVProcessor
FUNCTION __init__()
    SET self.traffic_data to None

FUNCTION load_csv_file(selected_date)
    CONSTRUCT file_path using selected_date
```

```
    TRY
        INITIALIZE traffic_data for junctions with 0 counts
        OPEN file at file_path
        READ file using CSV DictReader
        FOR each row in the file
            GET time_of_day from row
            IF time_of_day is valid
                EXTRACT hour from time_of_day
                GET junction_name from row
                IF junction_name is valid
                    INCREMENT vehicle count for the hour
        SET self.traffic_data to the populated traffic_data
        PRINT success message
    EXCEPT FileNotFoundError
        PRINT error message for file not found
    EXCEPT Exception as e
        PRINT error message for loading file

FUNCTION clear_previous_data()
    SET self.traffic_data to None
    PRINT message indicating data has been cleared

FUNCTION handle_user_interaction()
    WHILE True
        PRINT prompt for date input
        CALL validate_date_input() to get day, month, year
        CONSTRUCT selected_date from day, month, year
        CALL load_csv_file(selected_date)
        CALL process_csv_data(file_path) to get outcomes
```

```
        IF outcomes are valid
            CALL display_outcomes(outcomes)
            CALL save_results_to_file(outcomes)
            CREATE HistogramApp instance with self.traffic_data and formatted date
            CALL run() on the histogram app
            BREAK loop

    FUNCTION process_files()
        WHILE True
            CALL handle_user_interaction()
            PROMPT user if they want to process another file
            IF user input is "N"
                PRINT end of run message
                BREAK loop
            ELSE
                CALL clear_previous_data()

MAIN PROGRAM EXECUTION
    IF __name__ is "__main__"
        CREATE MultiCSVProcessor instance
        CALL process_files() on the instance
```