A close-up of a logo

Description automatically generated

Course : Computer Science Level 4

Module : 4COSC006C.1 Software Development I

Type of the assignment: Individual Course Work

Submission Date: 12.10.2024

|  |  |  |
| --- | --- | --- |
|  | Student ID | Student Name |
| IIT | 20232619 | S.H.S Yapa |
| UOW | W2121186 | S.H.S Yapa |

# Table of Contents

[Table of Contents 2](#_Toc184716615)

[Table of Figures 2](#_Toc184716616)

[Introduction 4](#_Toc184716617)

[Design 5](#_Toc184716618)

[Pseudo Code 5](#_Toc184716619)

[TEST CASES 8](#_Toc184716620)

[Positive Test Cases 8](#_Toc184716621)

[Negative Test Cases 9](#_Toc184716622)

[Screenshots 10](#_Toc184716623)

[References 13](#_Toc184716624)

# Table of Figures

[figure 1:check user inputs 9](#_Toc184716187)

[figure 2: Task B output1 (15/6/2024) 9](#_Toc184716188)

[figure 3: Task B output3 (21/6/2024) 10](#_Toc184716189)

[figure 4:Task B output2 (16/6/2024) 10](#_Toc184716190)

[figure 5:analyze another data set 11](#_Toc184716191)

[figure 6:file not found 11](#_Toc184716192)

[Table 1:Positive test cases 6](#_Toc184716395)

[Table 2: Negative Test Cases 7](#_Toc184716396)

# Introduction

This report documents a Python program designed to analyze traffic survey data. The program allows users to select a specific date, validates the input, and processes a CSV file containing traffic details. It calculates various statistics, such as the total number of vehicles, the percentage of trucks, and peak traffic hours. The results are displayed and saved to a text file for future reference. Additionally, the program ensures date accuracy by checking for leap years and handling invalid dates.

# 

# Design

## Pseudo Code

TRAFFIC\_ANALYSIS

START

FUNCTION validate\_date\_input

PROMPT "Please enter the day of the survey in the format dd"

GET day

WHILE day NOT IN range 1 to 31

DISPLAY "Out of Range - values must be in the range 1 and 31."

GET day

ENDWHILE

PROMPT "Please enter the month of the survey in the format MM"

GET month

WHILE month NOT IN range 1 to 12

DISPLAY "Out of Range - values must be in the range 1 and 12."

GET month

ENDWHILE

PROMPT "Please enter the year of the survey in the format YYYY"

GET year

WHILE year NOT IN range 2000 to 2024

DISPLAY "Out of Range - values must be in the range 2000 to 2024."

GET year

ENDWHILE

MAKE file\_name as "traffic\_data" + day + month + year + ".csv"

DISPLAY "Generated file name: file\_name"

TRY TO OPEN file\_name

IF file\_name EXISTS THEN

RETURN file\_name

ELSE

DISPLAY "File not found. Please try another date."

ENDIF

END FUNCTION

FUNCTION validate\_continue\_input

PROMPT "Do you want to analyze another dataset? (y/n):"

GET choice

WHILE choice NOT IN ["y", "n"]

DISPLAY "Invalid input. Enter 'y' or 'n'."

GET choice

ENDWHILE

RETURN choice == "y"

END FUNCTION

FUNCTION process\_csv\_data(file\_name)

INITIALIZE all result variables to 0

OPEN file\_name AS CSV

FOR EACH row IN CSV

CALCULATE required metrics

ENDFOR

RETURN metrics AS results\_list

END FUNCTION

FUNCTION display\_outcomes(results)

DISPLAY all metrics in results with explanations

END FUNCTION

FUNCTION save\_results\_to\_file(results, file\_name)

FORMAT results AS STRING

APPEND results STRING TO file\_name

DISPLAY "Results saved to file\_name"

END FUNCTION

MAIN PROGRAM

REPEAT

CALL validate\_date\_input()

STORE file\_name

IF file\_name EXISTS THEN

CALL process\_csv\_data(file\_name)

STORE results

CALL display\_outcomes(results)

CALL save\_results\_to\_file(results, "results.txt")

ELSE

DISPLAY "Invalid file name. Please try again."

ENDIF

CALL validate\_continue\_input()

CONTINUE\_PROGRAM

UNTIL CONTINUE\_PROGRAM IS FALSE

END

END

# TEST CASES

## Positive Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | The Scenario | Expected Result | Actual Result | Test Case “Pass” or “Fail” |
| 1 | Display user prompts | Prompts for date inputs (day,month,year) are displayed | Prompts are displayed correctly | Pass |
| 2 | Validate date inputs | User can only input valid day (1-31),month(1-12), year(2000-2024) | Inputs are validated correctly | Pass |
| 3 | Generate file name | File name is created using the date entered | File name matches the expected format | Pass |
| 4 | File existence  check | Program verifies if the file exists | File existence check works correctly | Pass |
| 5 | Process file data | Counts (e.g., total vehicles, trucks, bicycles) are calculated | Results are calculated correctly | Pass |
| 6 | Display results | Results (totals) are shown to the user | Results are displayed correctly | Pass |
| 7 | Save results to file | Results are saved to a file named “results.txt” | Results are saved successfully | Pass |
| 8 | User option to continue | User can choose whether to analyze another file or exit | Program correctly handles user input | Pass |

Table 1:Positive test cases

## Negative Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | The scenario | Expected Result | Actual Result | Test Case “Pass” or “Fail” |
| 1 | Validate day input (e.g., 32) | Reject invalid day input | Accepted invalid input | Fail |
| 2 | Reject non-numeric input for year | Rejects invalid year input (e.g., "abc") | Accepted invalid input | Fail |
| 3 | Process file data | Correctly processes the file and calculates totals | Totals are incorrect due to a logic error in code | Fail |
| 4 | User option to continue | Accepts only valid inputs (y or n) for continuation | Accepted an invalid input | Fail |

Table 2: Negative Test Cases

These are some of the negative test cases I identified. I have fixed the problems, and the program now works correctly in those situations.

# Screenshots

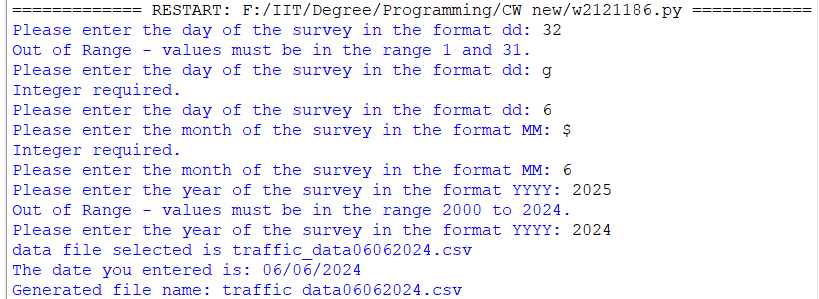


figure 1:check user inputs

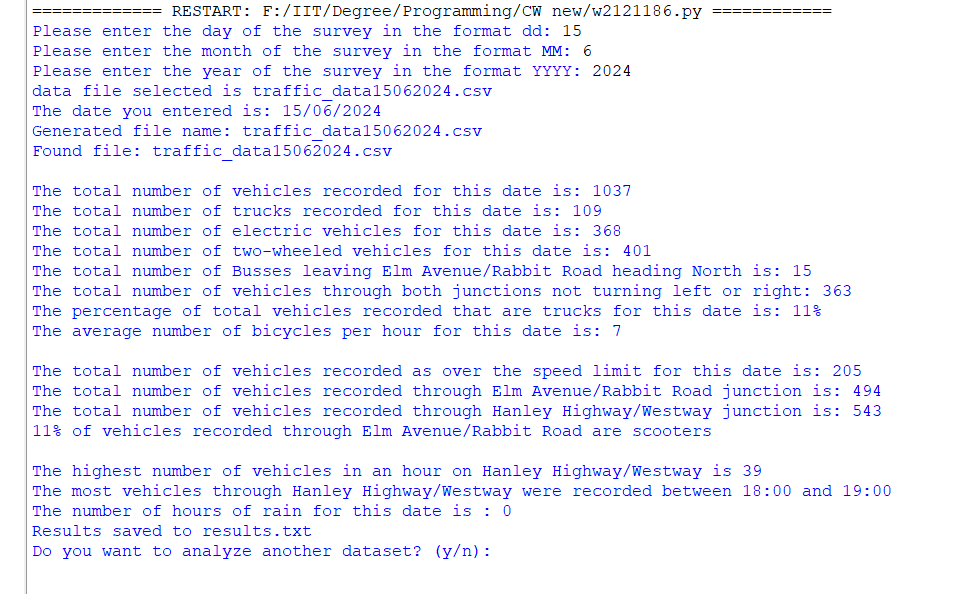


figure 2: Task B output1 (15/6/2024)

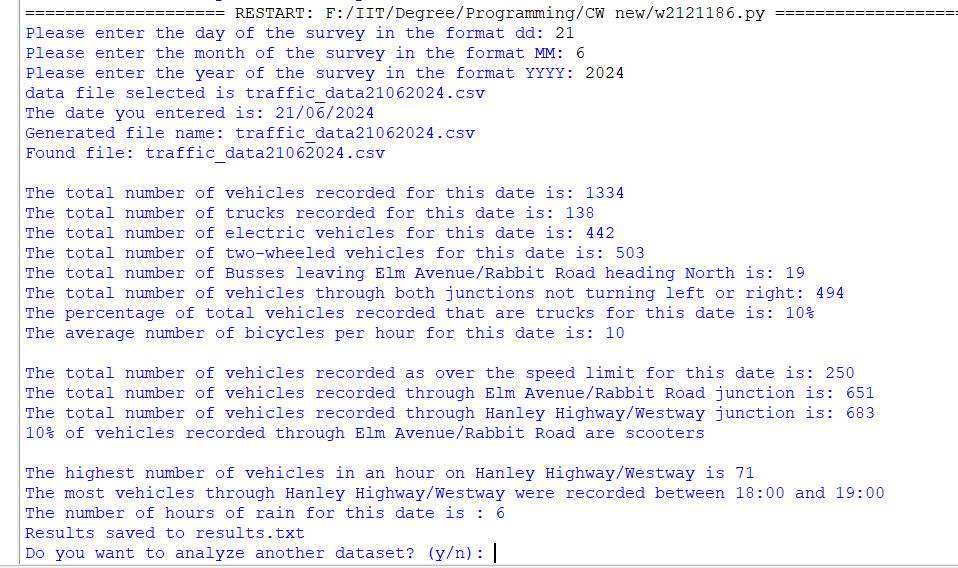


figure 3: Task B output3 (21/6/2024)

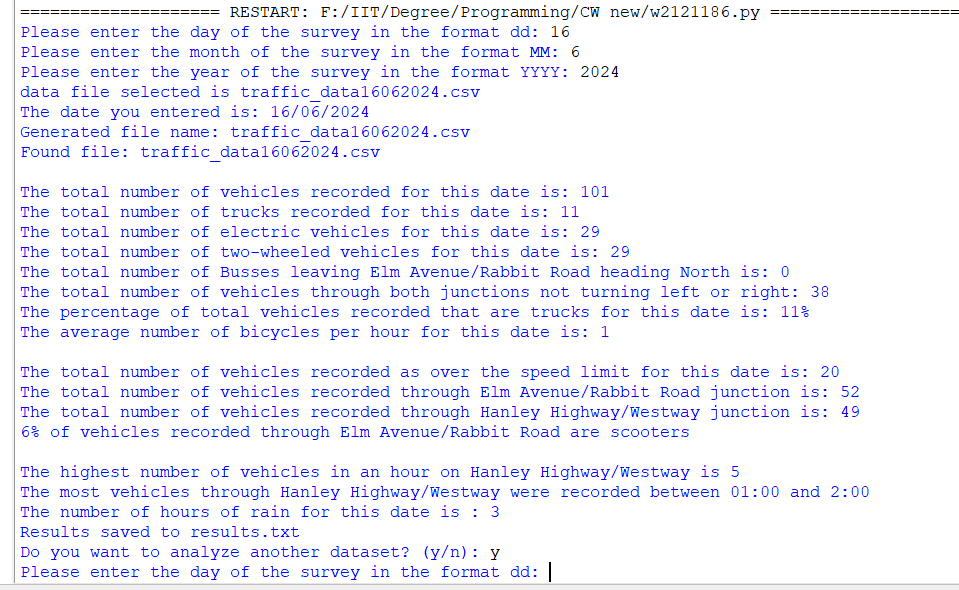


figure 4:Task B output2 (16/6/2024)

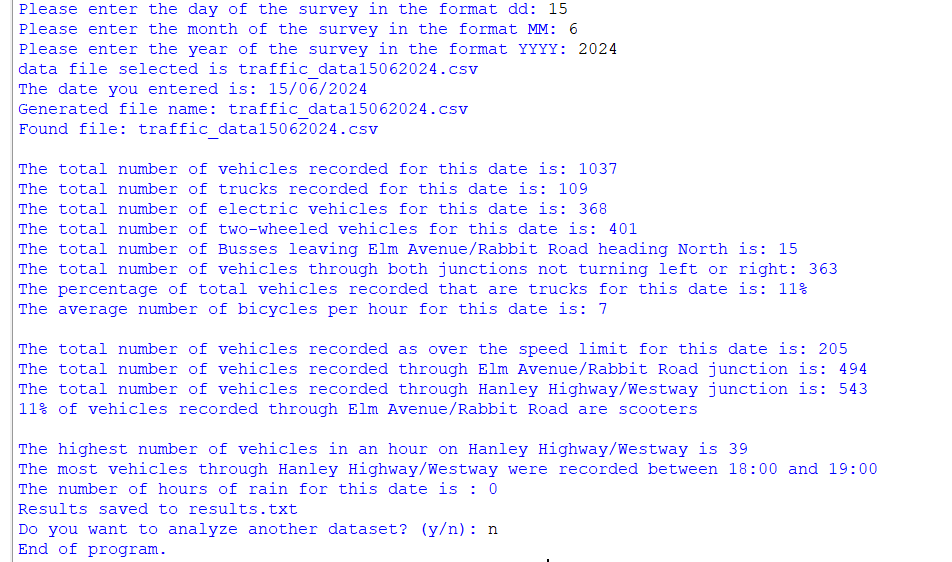


figure 5:analyze another data set

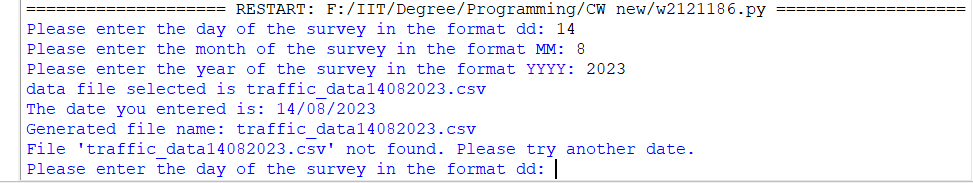


figure 6:file not found

# References

ChatGPT

W3 Schools - [Python Tutorial](https://www.w3schools.com/python/default.asp)