

**ASSIGNMENT 3**  
**PROGRAMMING TECHNIQUE 1 (SECJ1013)**  
**SECTION 04, SEM 1 (2024/2025)**

**INSTRUCTIONS TO THE STUDENTS**

- This assignment must be done ***in pairs***, except for students explicitly instructed to complete it individually.
- Please check your name in the ***Assignment 2 & 3 Groups List***:
- ***Program Requirements:***
  - ✓ Your program must strictly follow the input and output formats as specified in the question and examples. Test your program thoroughly using the given input examples and other possible test cases.
  - ✓ Plagiarism is strictly prohibited. Students who copy or share their work will receive ZERO marks (both the one who copies and the one who shares).
  - ✓ Programs detected as being 100% generated by AI tools will also receive ZERO marks.
  - ✓ Include the following details in the comments section of your program:
    - Your name and your partner's name (if applicable)
    - Matric number(s)
    - Date of completion of the assignment.

**SUBMISSION PROCEDURE**

- Submit the assignment ***before Thursday, January 23, 2025, at 12:00 AM***.
- Only one submission per group is required, which includes two type of files: the source code (the file with the extension *.cpp*) and the input files (the files with the extension *.txt*).
- Submit your assignment via the UTM e-learning system.
- Your submission will be evaluated based on correctness, clarity, formatting, and adherence to the requirements.

**QUESTION**

The Tourism Department of Malaysia tracks revenue and visitor data for popular tourist destinations across the country. Each destination is identified by a unique code (e.g., LGH101 for Langkawi Geopark, GNT202 for Genting Highlands). Monthly data on visitor count, ticket revenue, and expenditure on maintenance is recorded. Your task is to create a system that analyzes the data and generates detailed reports for each destination. This assignment assesses your understanding of arrays, functions, control structures, file operations, output formatting, and structured data. You are required to write a complete C++ program to address the problem. The program must perform the following tasks:

**Task 1: Define a Structured Data Type**

Define a structure to store the following data for each destination:

- (a) Destination code.
- (b) Destination name.
- (c) State.

- (d) Monthly data: Visitor count (January to December), Ticket revenue (RM) (January to December), and Maintenance cost (RM) (January to December).
- (e) Annual totals (calculated later for visitor count, revenue, and maintenance cost).

## Task 2: Read and Validate Input Files

Write a function that:

- (a) Checks if the following input files exist:
- destinations.txt: Contains destination codes, one per line.
  - visitors.txt, revenue.txt, maintenance.txt: Contain monthly data for visitors, ticket revenue, and maintenance cost, respectively.
- (b) Verifies that all files have the same number of lines to ensure data consistency.
- (c) Terminates the program with an appropriate error message if any file is missing or inconsistent.
- (d) Read data from the input files and store it in an array of Tourist Destination structures.
- (e) Dynamically determine the number of tourist destinations based on the number of lines in the files.
- (f) Figure 1 shows an example of data in the input files.

LGH101	10000	12000	11000	15000	14000	13000	12500	13500	14500	14000	15000	15500
GNT202	8000	9000	8500	12000	11000	10500	10000	11500	12500	12000	13000	13500
MEL303	12000	13000	12500	14000	13500	12800	13000	14000	14500	14200	15000	15500
PEN404	11000	12500	12000	13000	12500	12200	11800	12800	13200	13000	13500	13800
KKF505	9000	10000	9500	11000	10500	10200	10000	11500	12000	11800	12500	13000
TDC606	8500	9500	9000	10500	10000	9800	9500	10800	11500	11200	12000	12500
KTR707	7000	8000	7500	9000	8500	8300	8100	9000	9500	9200	10000	10500
<b>destinations.txt</b>	<b>visitors.txt</b>											

  

100000	120000	110000	150000	140000	130000	125000	135000	145000	140000	150000	155000
80000	90000	85000	120000	110000	105000	100000	115000	125000	120000	130000	135000
120000	130000	125000	140000	135000	128000	130000	140000	145000	142000	150000	155000
110000	125000	120000	130000	125000	122000	118000	128000	132000	130000	135000	138000
90000	100000	95000	110000	105000	102000	100000	115000	120000	118000	125000	130000
85000	95000	90000	105000	100000	98000	95000	108000	115000	112000	120000	125000
70000	80000	75000	90000	85000	83000	81000	90000	95000	92000	100000	105000
<b>revenue.txt</b>											

  

50000	60000	55000	70000	65000	63000	62000	67000	72000	71000	75000	77000
40000	45000	43000	60000	55000	53000	50000	58000	63000	61000	67000	69000
60000	65000	63000	70000	68000	65000	64000	69000	73000	72000	75000	77000
55000	62000	60000	65000	62000	61000	58000	64000	68000	67000	71000	73000
45000	50000	48000	55000	52000	51000	50000	57000	60000	59000	62000	64000
43000	48000	46000	52000	50000	49000	47000	54000	58000	56000	60000	62000
38000	42000	40000	45000	43000	42000	41000	45000	47000	46000	50000	52000
<b>maintenance.txt</b>											

Figure 1: Sample data in the input files

## Task 3: Determine Tourist Destination Details

Write a function that:

- (a) Maps each destination code to its corresponding name and state using the table below:

Code Prefix	Destination Name	State
LGH	Langkawi Geopark	Kedah
GNT	Genting Highlands	Pahang
MEL	Melaka Historical City	Melaka
PEN	Penang Hill	Penang
KKF	Kinabalu National Park	Sabah
TDC	Tasik Chini	Pahang
KTR	Kuala Terengganu Beach	Terengganu

(b) Updates the array of Tourist Destination structures with the destination name and state.

#### Task 4: Perform Data Analysis

Write a function that:

- Computes the annual totals for visitor count, ticket revenue, and maintenance cost for each destination.
- Calculates the net revenue for each destination using the formula:  

$$\text{Net Revenue} = \text{Annual Ticket Revenue} - \text{Annual Maintenance Cost}$$
- Computes the total visitors, revenue, and maintenance costs for each month across all destinations.
- Calculates the average revenue per visitor for each destination.
- Calculates the maintenance cost per visitor for each destination.

#### Task 5: Generate a Tourism Report

Write a function that:

- Displays the following data in a formatted table for each destination: Destination code, name, state, annual totals for visitors, revenue, maintenance costs, net revenue, average revenue per visitor, and maintenance cost per visitor.
- Displays the following summary statistics:
  - The total visitor count across all destinations.
  - The total ticket revenue across all destinations.
  - The total maintenance costs across all destinations.
  - The month with highest visitors and its value.
  - The month with lowest visitors and its value.
  - The destination with highest net revenue and its value.
  - The destination with lowest net revenue and its value.
- Figure 2 provides an example of the output displayed on the screen, based on the data from the input files illustrated in Figure 1.
- Ensure the output is neatly formatted and easy to read.

CODE	NAME	STATE	VISITORS	REVENUE (RM)	MAINT COST (RM)	NET REV (RM)	AVG REV/VIS	MAINT/VIS (RM)
LGH101	Langkawi Geopark	Kedah	160000	1600000	787000	813000	10.00	4.92
GNT202	Genting Highlands	Pahang	131500	1315000	664000	651000	10.00	5.05
MEL303	Melaka Historical City	Melaka	164000	1640000	821000	819000	10.00	5.01
PEN404	Penang Hill	Penang	151300	1513000	766000	747000	10.00	5.06
KKF505	Kinabalu National Park	Sabah	131000	1310000	653000	657000	10.00	4.98
TDC606	Tasik Chini	Pahang	124800	1248000	625000	623000	10.00	5.01
KTR707	Kuala Terengganu Beach	Terengganu	104600	1046000	531000	515000	10.00	5.08
-----								
Total Visitors: 967200								
Total Revenue: RM9672000.00								

```
Total Maintenance Costs: RM4847000.00
Month with Highest Visitors: Month 12 (94300 visitors)
Month with Lowest Visitors : Month 1 (65500 visitors)
Destination with Highest Net Revenue: Melaka Historical City (Melaka)
Destination with Lowest Net Revenue : Kuala Terengganu Beach (Terengganu)
-----
```

**Figure 2:** Example of output

***Additional Notes:*** Please include meaningful comments in your code for clarity.