

SULIT



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Fakulti
Komputeran

**UNIVERSITI TEKNOLOGI MALAYSIA
FINAL EXAMINATION (PRACTICAL)
SEMESTER 1, 2023/2024**

SUBJECT CODE : **SECJ1013**

SUBJECT NAME : **PROGRAMMING TECHNIQUE I**

SECTION : **01,02,03,04,05,07,08,09,10,11,20**

TIME : **(3 HOURS)**

DATE/DAY :

VENUES :

INSTRUCTIONS TO THE STUDENTS:

- This exam consists of **2 (TWO)** questions with a total of 100 marks.
- **Answer all** questions.
- You have **THREE HOURS** to complete the exam, including downloading the exam materials and submitting your programs.
- Your programs must adhere to the input and output requirements specified in the text and demonstrated in the examples. You must test the programs with (but not limited to) all the input provided in the examples.
- All the **COMMENT STATEMENTS** in the submitted program **WILL NOT BE EVALUATED**.

EXAM MATERIALS:

- You are given a program file with errors (**FinalQ1.cpp**) for Question 1.
- You are given a program file (**FinalQ2.cpp**) and an input file (**InputQ2.txt**) for Question 2.
- Download the files from UTM's e-learning.
- The provided program files should be used as the base for answering the questions in this exam.

SUBMISSION PROCEDURE:

- Only program file (with the extension .cpp) are required for the submission.
- Submit the program file (with the extension .cpp) through UTM's e-learning.

This question booklet consists of **SIX (6)** printed pages excluding this page.

QUESTION 1 - ERROR DEBUGGING

[35 MARKS]

Figure 1 shows the C++ program (**FinalQ1.cpp**) that is created using function, array, pointer and file. The main function reads an integer number *n* (more than 0 and less than 25) from user input and then reads *n* integers from a file (called *input.dat*). The file data is assumed to contain at least *n* values. If the number that is read from user input is within range, the program will display an output as shown in **Figure 2**. However, if the number that is read from user input is out of range, the program displays no output as shown in **Figure 3**. The details of the function are described as below:

- A function named as `anotherArray` takes an `int` array and its size as arguments. The new array that the function creates triple size as large as the argument array.
- The function copies every element of the argument array to the new array. The function initializes the unused members of the new array with 0.
- After passing the array to function, the program prints one value per line on standard output representing the contents of the new array.
- A pointer of the new array is returned by the function.

There are numbers of syntax or logical errors found in the program (**FinalQ1.cpp**). You are required to debug the errors, compile, and run the program. You are **NOT ALLOWED** to remove any statements from the program. You are only **ALLOWED** to update the statements provided in the program and add a new statement(s) if necessary.

```
1  #include <iostream>
2  #include <ofstream>
3
4  using namespace std;
5
6  // Function Prototype
7  int *anotherArray(int[], int);
8  void displayArray(int[], int);
9
10 //Function Main
11 int main()
12 {
13     // Declaration
14     const int SIZE = 25;
15     int values[size];
16     int n;
17
18     // The Input Data
19     cin >> n;
```

```

20     if (n<0 && n>=SIZE)
21         return 0;
22
23     // File Processing
24     ifstream inFile;
25     inFile.open("input");
26
27     for (int k=0; k<n; k++)
28         inFile >> values[n]; // Reading from File
29     inFile.closed();
30
31     // Declare and create another array
32     int *arrCopy = anotherArray(values, SIZE);
33
34     // Display the contents of the new array.
35     displayArray(arrCopy, n*2);
36
37     return 0;
38 }
39
40 //Create Another Function
41 int &anotherArray(int arr[], int size)
42 {
43     // Verify the size value
44     if (size <= 0)
45         return NULL;
46
47     // Declare and create triple size of a new array
48     int *copy = new int[size];
49
50     // Copy the value to the new array
51     for (int index = 1; index < size; index++)
52         copy[index] = arr[index];
53
54     // Assign the remaining elements to 0
55     for (int index = 1; index < size * 3; index++)
56         copy[index] = 0;
57
58     // Return a pointer to the new array.
59     return copy[size];
60 }
61
62 //Create Display Function
63 void displayArray(int arr[], int size)
64 {
65     for (int index = 0; index < size; index++)
66         cout << arr[index] << endl;
67
68     return 0;
69 }

```

Figure 1: C++ program code for FinalQ1.cpp

```
4 [Enter]
4929104
0
32
0
0
0
0
0
0
Process exited after 12.63 seconds with return value 0
Press any key to continue ...
```

Figure 2: C++ program example output 1

```
26 [Enter]
Process exited after 12.63 seconds with return value 0
Press any key to continue ...
```

Figure 3: C++ program example output 2

QUESTION 2 - PROBLEM SOLVING

[65 MARKS]

Malaysia Automotive Association (MAA) generates an annual report on sales and production statistic report to ascertain average of sales, total and percentage, lowest and highest brand that has been sold. You are required to write a complete C++ program based on a statistic report by brand from year 2016 to 2019. Answer the question using the provided template file (**FinalQ2.cpp**) and input file (**InputQ2.txt**). **Figure 4** shows the complete definition of the `displayLine` function. The purpose of the other functions is as the name implies and further explained below. Write the C++ program to perform the following tasks:

```
void showLine()
{
    for (int i = 0; i < 60; i++) {
        cout << "-";
        cout << endl;
    }
}
```

Figure 4: Function definition for `displayLine`

Task 1: Declare a structure named `Sales` with the following members: (4 marks)

- a) A brand name.
- b) A sales percentage for each brand from 2016 to 2019. You should use a one-dimensional array to store the percentage for four years.
- c) A four-year average of the sales percentage.

Task 2: Define a function named `calculateAverage`. (7.5 marks)

- a) This is a non-returning function.
- b) It receives as input parameter an array of the structure defined in Task 1.
- c) Calculate the four-year average of sales percentage for each brand. The average calculation result will be saved in the average member variable of the structure defined in Task 1.

Task 3: Define a function named `findLowest`. (9 marks)

- a) It receives an array of the structure defined in Task 1 as input parameter.
- b) Determine the brand with the lowest four-year average of sales percentage by identifying the index of the array received in 3(a) with the lowest four-year average of sales percentage.
- c) The function should return the array index identified in 3(b).

Task 4: Define a function named `findHighest`. (9 marks)

- a) It receives an array of the structure defined in Task 1 as input parameter.
- b) Determine the brand with the highest four-year average of sales percentage by identifying the index of the array received in 4(a) with the highest four-year average of sales percentage.
- c) The function should return the array index identified in 4(b).

Task 5: Define a function named `displayOutput`. (19 marks)

- a) This is a non-returning function.
- b) It receives as input parameter an array of the structure defined in Task 1.
- c) The function should show the brand, the sales percentage from 2016 to 2019, the four-year average of sales percentage, and the brand with the lowest and highest four-year average of sales percentage. You should use the `displayLine` function as well as the functions defined in Tasks 3 and 4. Please use proper output formatting. **Figure 5** shows the output that will be displayed on the screen based on the data in the input file (**InputQ2.txt**) which is shown in **Figure 6**.

Task 6: Define a main function to perform the following tasks: (16.5 marks)

- a) Declare **one-dimensional array** variable for the structure defined in Task 1.
- b) The function should read data from the input file (**InputQ2.txt**). The file contains the statistic of sales percentage for each brand from 2016 to 2019. Please ensure that the program only continues reading the file if it is successfully opened; otherwise, print the error message and exit the program.
- c) The function must invoke the functions defined in Task 2 and Task 5 in order to produce the output shown in **Figure 5**.
- d) Finally, include all necessary header files.

Percentage of Sales by Brand (2016 to 2019)					
BRAND	2016	2017	2018	2019	AVERAGE
Honda	17.8	21.3	20.2	15.6	18.73
BMW	2.3	2.4	2.3	1.8	2.20
Mercedes	2.4	2.3	2.5	1.8	2.25
Perodua	40.3	39.8	42.6	40.8	40.88
Kia	1.2	1.0	1.1	1.4	1.18
Mazda	1.7	2.2	3.0	2.9	2.45
Nissan	6.5	5.9	5.3	4.8	5.62
Proton	17.8	14.2	13.1	17.6	15.67
Volkswagen	1.3	1.5	1.3	0.9	1.25
Toyota	8.7	9.4	8.6	12.4	9.77
Lowest Average of Sales Percentage: 1.18 - Kia Highest Average of Sales Percentage: 40.88 - Perodua					

Figure 5: Output of the program

17.8	21.3	20.2	15.6	Honda
2.3	2.4	2.3	1.8	BMW
2.4	2.3	2.5	1.8	Mercedes
40.3	39.8	42.6	40.8	Perodua
1.2	1.0	1.1	1.4	Kia
1.7	2.2	3.0	2.9	Mazda
6.5	5.9	5.3	4.8	Nissan
17.8	14.2	13.1	17.6	Proton
1.3	1.5	1.3	0.9	Volkswagen
8.7	9.4	8.6	12.4	Toyota

Figure 6: Statistic of sales in the file InputQ2.txt