UNIVERSITI
TEKNOLOGI
PETRONAS

FINAL EXAMINATION
MAY 2024 SEMESTER

COURSE	**:**	**TEB1013/TFB1013 - STRUCTURED PROGRAMMING**
DATE	**:**	**8 AUGUST 2024 (THURSDAY)**
TIME	**:**	**2:30 PM - 5:30 PM (3 HOURS)**

INSTRUCTIONS TO CANDIDATES

1. Answer **ALL** questions in the Answer Booklet.
2. Begin **EACH** answer on a new page in the Answer Booklet.
3. Indicate clearly answers that are cancelled, if any.
4. Where applicable, show clearly steps taken in arriving at the solutions and indicate **ALL** assumptions, if any.
5. **DO NOT** open this Question Booklet until instructed.

Note :

i. There are **NINE (9)** pages in this Question Booklet including the cover page

ii. **DOUBLE-SIDED** Question Booklet.

1. Trace the output and find the appropriate answer for each of the following program segments.

a.

```
int a = 8, b = 4, c = 15, d = 2, e = 21;
cout << a * b + c - d % e / b << endl;
cout << (a + c) * e / d << endl;
cout << a - (b * c % d) + e * d << endl;
```

FIGURE Q1a: Program segment 1a

[5 Marks]

b.

```
string userInput;
cout << "Enter a string: ";
cin >> userInput;

if (userInput == "structured" || userInput == "programming")
{
   cout << "You entered: " << userInput << endl;

   int year;
   cout << "Enter a year to check if it's a leap year: ";
   cin >> year;

   if ((year % 4 == 0 && year % 100 != 0) || (year % 400 ==
0)) {
      cout << year << " is a leap year" << endl;
        } else {
      cout << year << " is not a leap year" << endl;
        }
    } else {
      cout << "You did not enter 'structured programming'.
Leap year check skipped." << endl;
}
```

FIGURE Q1b: Program segment 1b

[5 Marks]

c.

```
int numbers[10] = {7, 8, 6, 3, 2, 9, 1, 5, 4, 0};
    for(int i = 1; i <= 4; i++) {
        int k = i * 2;
        int val1 = numbers[i];
        int val2 = numbers[k];
        int val3 = numbers[k + 1];
        int val4 = numbers[9 - i];
        int sum = val1 + val2 + val3 + val4;

        cout << "numbers[" << i << "] = " << val1 << ", ";
        cout << "numbers[" << k << "] = " << val2 << ", ";
        cout << "numbers[" << k + 1 << "] = " << val3 << ", ";
        cout << "numbers[" << 9 - i << "] = " << val4 << ", ";
        cout << "Sum = " << sum << endl;
    }
    return 0;
        }
```

FIGURE Q1c: Program segment 1c

[5 Marks]

d.

```
typedef struct {
    string name;
    int score;
} PLAYER;

int main() {
    PLAYER players[3];
    players[2] = {"Alice", 85};
    for (int i = 0; i < 2; i++) {
        cout << players[i].name << "score: " <<
players[i].score << endl;
    }
```

FIGURE Q1d: Program segment 1d

[5 Marks]

2. a. Write a C++ program that should use the concept of function overloading and creates **TWO (2)** functions to produce the output in **FIGURE Q2a**. Assume the following values for each parameter:

- Rectangle: length = 10.0, width = 5.0
- Triangle: base = 8.0, height = 5.0

```
Area of the rectangle: 50
Area of the triangle: 20
```

FIGURE Q2a: Output of a C++ program for functions

[5 Marks]

b. Write a C++ program using nested for loops to generate a half-pyramid of alternating * and A characters as shown in **FIGURE Q2b**. The number of rows in the pyramid should be entered by the user.

```
*
* A
* A *
* A * A
* A * A *
```

FIGURE Q2b: Output of a C++ for nested for loops

[10 Marks]

c. Write a C++ program that should demonstrate the use of loops to manipulate arrays and generate the specified output shown in **FIGURE Q2c.**

```
AIMHIGH!
```

FIGURE Q2c: Output of a C++ program having loops and arrays

[5 Marks]

3. a. Write a C++ program that allows a user to enter details for 5 items. For each item, prompt the user to input the name (as a single word without spaces), price, and quantity purchased. Calculate and display the total cost of all items purchased. A sample output is shown in **FIGURE Q3a**.

```
Enter details for 5 items:
Item 1 Name: Shampoo
Item 1 Price: 5
Item 1 Quantity: 1
Item 2 Name: Soup
Item 2 Price: 3
Item 2 Quantity: 3
Item 3 Name: ToothPaste
Item 3 Price: 3
Item 3 Quantity: 2
Item 4 Name: Conditioner
Item 4 Price: 6
Item 4 Quantity: 1
Item 5 Name: FaceWash
Item 5 Price: 4
Item 5 Quantity: 2
Total Cost of all items: $34
```

FIGURE Q3a: Sample output to calculate and display the total cost of purchased items

[10 Marks]

b. Extend the program in **part (a)** to display the quantity of each item purchased after calculating the total cost as shown in **FIGURE Q3b**. Ensure that each item's details (name, price, quantity) are stored using arrays.

```
Quantity of items purchased:
Shampoo: 1 items
Soup: 3 items
ToothPaste: 2 items
Conditioner: 1 items
FaceWash: 2 items
```

FIGURE Q3b: Sample output to display the quantity of each item purchased

[10 Marks]

4. A new private college has recently commenced operations, and while their marking and grading system is currently manual, the Registration and Examination department is aiming to implement an automated system to calculate students' final grades based on their coursework and exam marks.

The coursework marks (100 marks) are divided into four categories: Lab (20 marks), Quiz (20 marks), Test (20 marks), and Project (40 marks). The final exam is marked out of 100, with the exam contributing 50% to the total grade. The remaining 50% is contributed by the coursework marks. The passing threshold for the course is set at 40%. Students scoring below 40% will fail the course. The grading scale is structured as follows:

TABLE Q4: Student grading system policy

Student Course Marks Percentage (%)	Student Grades
80-100%	A
70-79%	B
50-69%	C
40-49%	D
Below 40%	F

a. Create a **structure** named `StudentMarks` with the following members:

- lab (an integer representing marks out of 20)
- quiz (an integer representing marks out of 20)
- project (an integer representing marks out of 60)
- finalexam (an integer representing marks out of 100)

[5 Marks]

b. Create a function named `calculateFinalPercentage` that takes a `StudentMarks` structure as input and returns the final percentage, considering that the coursework contributes 50% of the total grade and the final exam contributes the other 50%. To do this:

- Calculate the total coursework marks out of 100.
- Convert the coursework marks to a percentage of the total grade (50%).
- Convert the final exam marks to a percentage of the total grade (50%).
- Get the final percentage.

[5 Marks]

c. Create a function named `determineGrade` that takes the final percentage as input and returns the corresponding grade based on the specified ranges shown in **TABLE Q4**.

[5 Marks]

d. Define a `main()` function that:

- Prompt the user to enter the marks for lab, quiz, project, and final exam marks.
- Calculate the final percentage using the `calculateFinalPercentage` function.
- Determine the grade using the `determineGrade` function.
- Display the calculated percentage and the grade.

[5 Marks]

5. a. Explain the program design process and draw a clear flowchart showing all the stages.

[6 Marks]

b. Define the difference between the following terms associated with functions:

- Actual vs. Formal Parameter,
- Value vs. Reference Parameter.

[4 Marks]

c. Write a C++ program that reverses an integer array `arr` of size 5 (e.g. `arr[5]`). Initialize the array `arr` with values {1, 2, 3, 4, 5}. After reversing the array, display the reversed array elements separated by spaces as shown in **FIGURE Q5c**.

```
Reversed array: 5 4 3 2 1
```

FIGURE Q5c: Sample output to display reverse array

[5 Marks]

d. Draw a flowchart of a C++ program for a quiz game. The program should repeatedly ask the user multiple-choice questions, evaluate the user's answers, and provide feedback. The quiz consists of 5 questions, each with 4 possible choices (A, B, C, D). After answering all questions, the program should calculate and display the user's score as a percentage.

[NOTE: Specify the type of loop used in the condition statement, such as while or do-while.]

[5 Marks]

- END OF PAPER -