UTHM
Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2023/2024

COURSE NAME	:	ALGORITHMS AND PROGRAMMING
COURSE CODE	:	BIC 10204
PROGRAMME CODE	:	BIP / BIW
EXAMINATION DATE	:	JULY 2024
DURATION	:	3 HOURS
INSTRUCTIONS	:	1. ANSWER ALL QUESTIONS 2. THIS FINAL EXAMINATION IS CONDUCTED VIA ☐ Open book ☒ Closed book 3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSED BOOK

THIS QUESTION PAPER CONSISTS OF **FIVE (5)** PAGES

Q1 Based on **Figure Q1.1**.

A university wants to automate the process of calculating students' final grades for a course. The course consists of several components: assignments, quizzes, a midterm exam, and a final exam. Each component contributes a certain percentage to the final grade. The university needs a program to input students' scores for each component, calculate their final grades based on the provided weights, and display the results. Data provided as below:

i. The weights of each component for the final grade calculation:
   Assignments: 30%
   Quizzes: 20%
   Midterm Exam: 25%
   Final Exam: 25%

ii. Scores of each student for the components:
   Assignments (out of 100)
   Quizzes (out of 50)
   Midterm Exam (out of 100)
   Final Exam (out of 100)

Figure Q1.1

(a) Determine appropriate Input, Processes and Output.

(5 marks)

(b) Draw a flow chart to solve the problem.

(15 marks)

Q2 Answer the following questions based on the given case study in **Figure Q2.1**.

You've been tasked with developing a program to calculate the final grade of students based on their scores in different subjects. The grading system is as follows:

- Scores above or equal to 90% are awarded an 'A' grade.
- Scores between 80% and 89% are awarded a 'B' grade.
- Scores between 70% and 79% are awarded a 'C' grade.
- Scores between 60% and 69% are awarded a 'D' grade.
- Scores below 60% are awarded an 'F' grade.

Requirements:

1. The program should prompt the user to enter the scores for different subjects (e.g., Mathematics, Science, English).
2. Calculate the average score of the student.
3. Based on the average score, determine and display the corresponding grade using if...else...if statements.

Figure Q2.1

(a) Write a complete pseudo-code to perform the calculation.

(10 marks)

(b) Write a complete C program that permits interaction with the user in this manner, see **Figure Q2.2**.

```
Enter the score for Mathematics: 85
Enter the score for Science: 78
Enter the score for English: 92
Average score: 85.00
Final grade: B
```

Figure Q2.2

(15 marks)

Q3 Answer the following questions based on the given case study in **Figure Q3.1**.

```
You've been tasked with creating a simple program that
converts temperatures between Celsius and Fahrenheit
scales. The program should allow users to input a
temperature value along with the scale (C for Celsius,
F for Fahrenheit) and then display the converted
temperature.

Requirements:
1. The program should prompt the user to enter a
   temperature value and the scale (C for Celsius, F for
   Fahrenheit).
2. Perform the conversion based on the user's input:
   - If the scale is Celsius (C), convert the
     temperature to Fahrenheit using the formula: F =
     (C × 9/5) + 32.
   - If the scale is Fahrenheit (F), convert the
     temperature to Celsius using the formula: C = (F
     - 32) × 5/9.
3. Display the converted temperature.
```

Figure Q3.1

Write a complete C program that implements the above requirements using the function statement to handle different temperature scales. Ensure that the program provides appropriate feedback for invalid inputs (e.g., unrecognized scales). The functions need to be involved including such below.

i. `float getTemperature();`

ii. `float convertTemperature(float temperature, char scale);`

iii. `void displayConvertedTemperature(float converted_temperature, char scale);`

(20 marks)

Q4 Answer the following questions based on the given case study in **Figure Q4.1**.

```
You've been assigned to create a simple ATM simulator
program that mimics the functionality of an Automated
Teller Machine (ATM). The program should allow users to
perform basic banking transactions such as checking
balance, depositing money, and withdrawing money.
Requirements:

1. The program should provide a menu-driven interface
   for users with the following options:
   -  Check balance
   -  Deposit money
   -  Withdraw money
   -  Exit
2. The program should maintain a user's account balance.
3. Users should be able to check their balance (an
   assumption for initial balance is RM1000), deposit
   money into their account, and withdraw money from
   their account.
4. The program should use the switch...case statement to
   handle different menu options.
```

Figure Q4.1

Write a complete C program that implements the above requirements for a simple ATM simulator using `switch...case` statements. Ensure that the program provides appropriate feedback for invalid inputs (e.g., insufficient balance for withdrawal). The sample of program output is shown in **Figure Q4.2**.

```
Simple ATM Simulator
1. Check balance
2. Deposit money
3. Withdraw money
4. Exit
Enter your choice: 1
```

Figure Q4.2

(20 marks)

Q5 Based on **Figure Q5.1**.

```
for (int i = 1; i <= 5; i++)
{
      printf("%d\n", i);
}
```

Figure Q5.1

Explain the purpose and step by step operation of a `for-loop` statement in C programming with the use of stated example.

(10 marks)

Q6 Based on **Figure Q6.1**, determine the output.

```
#include <stdio.h>
int main()
{
int numbers[5] = {10, 20, 30, 40, 50}; //Static initialization

//Access and print array elements
      printf("Array elements: \n");

      printf("numbers[0]: %d\n", numbers[0]);
      printf("numbers[1]: %d\n", numbers[1]);
      printf("numbers[2]: %d\n", numbers[2]);
      printf("numbers[3]: %d\n", numbers[3]);
      printf("numbers[4]: %d\n", numbers[4]);
      return 0;
}
```

Figure Q6.1

(5 marks)

- END OF QUESTIONS -