BSc. Engg. CSE 1st Semester August 27, 2021

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **WINTER SEMESTER, 2020-2021**
DURATION: 1 Hour 30 Minutes **FULL MARKS: 75**

CSE 4107: Structured Programming I

*There are **3 (three)** questions. Answer all **3 (three)** of them. Figures inside the boxes in right margin indicate marks of each question. The blue square brackets on the start of each question denotes the corresponding CO(s) and PO(s). The name of the pdf must be in the following format <**StudentID CourseCode MID**>.*

---

1. (a) [CO1, CO2, PO1] *Summarize* **3 control structures** of structured programming. For each of the control structures, the summary should **not contain more than 4 sentences** and give **1 example written in C** for each of the structures. 9

   (b) [CO2, CO4, PO2, PO4] *Break down* the following snippet of C code into **tokens**. For each of the tokens, **mention it’s type**. 12

```
const int DAYS_IN_A_WEEK = 7;
printf("There are %d days in a week.\n", DAYS_IN_A_WEEK);
return 0;
```

   (c) [CO1, PO1] *Explain* under **10 simple sentences** why using expressive and proper identifiers along with proper coding conventions is important. 4

Total for Question 1: 25

2. (a) [CO3, CO4, PO4] In the following code, *change* the: 5

   1. `while` loop to equivalent `for` loop
   2. `getchar` statement to equivalent `scanf` statement
   3. `switch` case to equivalent `if ... else` statement
   4. `putchar` statements to equivalent `printf` statements

```
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    char ch;

    while((ch = getchar()) != ’\n’) // Change while loop to for loop, char scanning from getchar to scanf
    {
        switch (ch) // switch to if...else
        {
        case ’a’: case ’e’: case ’i’: case ’o’: case ’u’:
            putchar(toupper(ch)); // putchar to printf
            break;
        default:
            putchar(tolower(ch)); // putchar to printf
            break;
        }
```

```c
    }

    return 0;
}
```

(b) [CO4, PO2, PO4] *Write* the output of the following code. 10

```c
#include <stdio.h>

int main(void)
{
    int i, j, k;

    for(i = 0, j = 10, k = 100; k < 406; i++, j++, k++)
    {
        printf("%3.3d, %3.3d, %3.3d\n", i, j, k);
        int temp = i;
        i = ++j * 2;
        j = k++ * 2;
        k = ++temp;
    }
    printf("%3.3d, %3.3d, %3.3d\n", i, j, k);
    return 0;
}
```

(c) [CO4, PO2, PO4] *Identify* any **mistake** or **error** in the following code. Mistakes and errors are those which will either break the logic of the code or might result in a compilation or run time error. For each error, write **one sentence** explaining what kind of error it is. 10

```c
#include <stdio.h>

int main(void)
{
    const double diagonalValue = 1.0;

    int n;
    double diagonalMultiplier;

    printf("Enter one dimension of the square matrix: ");
    scanf("%d", n);
    printf("Enter diagonal multiplier value: ");
    scanf("%f", &diagonalMultiplier);
    diagonalValue *= diagonalMultiplier;

    double mat[n][n]; // n x n matrix using VLA

    // The following loop sets the principal diagonal to be updated diagonal value
    for(i = 0; i <= n; i++)
    {
        for(j = 0; j <= n; j++)
        {
            if(i = j)
                mat[i][j] = diagonalValue;
        }
    }

    // Prints out the matrix
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < n; j++)
        {
            printf('[%f]\t', mat[i, j]);
        }
        printf("\n");
    }

    return 0;
}
```

Total for Question 2: 25

3. (a) [CO2, PO1] *Briefly* explain the `typedef` keyword in C along with an example. 5

(b) [CO3, PO2, PO3] Matrices can be represented using two dimensional arrays in C. The determinant of a 2 × 2 matrix can be written as, 10

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Similarly, for a 3 × 3 matrix, the determinant will be,

$$|A| = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a\begin{vmatrix} e & f \\ h & i \end{vmatrix} - b\begin{vmatrix} d & f \\ g & i \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

Write a program in C that will read a 3 × 3 matrix from the user and store it in a 2D array. Then *calculate* the **determinant** of the matrix and print it. Scan the matrix from the user in a **sensible way**.

(c) [CO3, CO5, PO1, PO2, PO3] A quadratic equation can be written as : 10

$$ax^2 + bx + c$$

Where, a, b and c are the coefficients. Given the coefficients of a quadratic equation, *write* a C program that would print the **equation**, **first order** and **second order derivatives** and **roots** of the equation. Use ^ symbol to represent the power. Match the inputs and outputs with the sample outputs. Handle signs between coefficients carefully. You can consider that only equations with real roots will be given.

The roots of a quadratic equation can be calculated using the following equation.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Sample Input:

```
Enter the co-efficients of quadratic equation:  -1.2 -5 3.2
```

Sample Output:

```
Given equation:  -1.2x^2 - 5x + 3.2
```

```
1st order derivative:  -2.4x - 5
2nd order derivative:  -2.4
Roots:  -4.7304, 0.56373
```

Total for Question 3: 25