B.Sc. Engg. EEE 2nd Semester 24 May 2024 (Morning) 4

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION SUMMER SEMESTER, 2022-2023
DURATION: 3 HOURS FULL MARKS: 100

CSE 4271: Computer Programming

Programmable calculators are not allowed. Do not write anything on the question paper.
Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions with corresponding COs and POs in parentheses.

1. a) Discuss the advantages and disadvantages of text and binary streams during file I/O. 5 (CO1) (PO1)

   b) Discuss the limitations of the switch statement. How are these limitations overcome in the if statement? 5 (CO1) (PO1)

   c) Write a program that takes an integer and prints whether it is positive, negative, or zero. The program also prints whether the number is even or odd. You cannot use the if condition anywhere within your code. You can only use the switch case. Some sample outputs for the problem are given in Table 1. 10 (CO3) (PO2)

Table 1: Sample output for Question 1.c

Sample Input	Sample Output
4	Positive and Even
-1	Negative and Odd
5	Positive and Odd

2. a) 'Macros are not variables' - Explain this statement with a suitable example. 5 (CO2) (PO1)

   b) Write the output of the code in Code Snippet 1. 5 (CO2) (PO1)

```
#include <stdio.h>
#define PI 3.1416
#define RAD 2
float area(){
    return PI*RAD*RAD;
}
int main (void){
    float num = area();
    printf("%.02f", num);
    return 0;
}
```

Code Snippet 1: C program for Question 2.a

3. a) Write a code that will read a binary file named *'myfirstfile'* and write the contents of that file in another binary file named *'mysecondfile'* in reverse order. You can assume the binary file only contains characters. 10 (CO3) (PO2)

   b) Write the output of the code in Code Snippet 2. 10 (CO2) (PO1)

```
#include<stdio.h>
int num[5] = {1, 2, 3, 4, 5};
void f2(int *p){
    int i;
    p = p + 5;
    for(i=0;i<5;i++)
        printf("%d ",*(--p));
    printf("\n");
    return;
}
void f1(int *p){
    int i;
    p++;
    for(i=0;i<4;i++)
        printf("%d ",p[i]);
    printf("\n");
    f2(num);
    return;
}
int main(){
    int num[5] = {11, 12, 13, 14, 15};
    char str[50] = "Hello World";
    char *p = str;
    char **q = &p;
    f1(num);
    while(**q){
        printf("%c", *p);
        p++;
    }
    return 0;
}
```

Code Snippet 2: C program for Question 3.b

4. a) Differentiate between pass by value and pass by reference using suitable examples. 7 (CO1) (PO1)

   b) Write a recursive function *int natsum(int num)* that returns the sum of all the natural numbers up to *num*, inclusive. Natural numbers are all positive integers from 1 to infinity. 10 (CO3) (PO2)

5. a) Write a program that determines if a square integer matrix is upper-triangular, lower-triangular, both, or none. A matrix in which all the elements below the main diagonal are zero is called an upper-triangular matrix. A matrix in which all the elements above the main diagonal are zero is called a lower-triangular matrix. Your program will first prompt the user to enter the length $l(2 \leq l \leq 100)$ of the matrix. It will then take $l \times l$ elements of the matrix. The program will then print the correct output. Some sample outputs for the problem are given in Table 2. 12 (CO3) (PO2)

Table 2: Sample output for Question 5.a

Sample Input	Sample Output
Enter length: 4 4 0 0 0 9 7 0 0 0 5 5 0 1 0 0 6	Lower Traingular
Enter length: 4 4 0 5 2 9 7 0 8 0 5 5 0 1 0 0 6	None

b) Differentiate between local and global variables using suitable examples. 5 (CO1) (PO1)

6. a) Write a program that will take an integer, *len* as input and print an hourglass pattern whose length will depend on *len*. A sample output for $len = 4$ the problem is given in Table 3. Note that there are spaces between the asterisks (*) in the output. 8 (CO3) (PO2)

Table 3: Sample output for Question 6.a

Sample Input	Sample Output
Enter length: 4	* * * * * * * * * * * * * * * * * * *

b) Write a program that takes two words as input and checks if they are anagrams of each other or not. An anagram is a word that is formed by rearranging the letters of another word. You can assume the words will consist a maximum of 100 characters. Some sample outputs for the problem are given in Table 4. 8 (CO3) (PO2)

Table 4: Sample output for Question 6.b

Sample Input	Sample Output
Enter word: dusty Enter word: study	Anagram
Enter word: peach Enter word: cheap	Anagram
Enter word: egg Enter word: gap	Not anagram