CS2063 Data Structures and Algorithms II

Lab Work 1 – Recursion

Q1. In the triangle.java program, remove the code for the base case

```
public static int  triangle(int n)
{
    if(n==1)
        return 1;
    else
        return( n + triangle(n-1) );
}
```

Remove this part

Then run the program and see what happens.

Q2. Write a recursive method to find the nth Fibonacci number.

Fn = Fn-1 + Fn-2

Where Fn is the nth term or number Fn-1 is the (n-1)th term Fn-2 is the (n-2)th term

Example: Fib(6) = 8 (since the Fibonacci sequence is 0, 1, 1, 2, 3, 5, 8, …)

Q3. Write a recursive method that computes the sum of all numbers from 1 to n, where n is given as parameter.

Q4. Write a recursive method that finds and returns the minimum element in an array, where the array and its size are given as parameters.

Q5. Write a recursive method that computes and returns the sum of all elements in an array, where the array and its size are given as parameters.

Q6. Write a Java program to reverse a string using recursion.

Q7. Imagine you are organizing a single-elimination sports tournament (e.g., tennis, soccer, etc.). In such a tournament, teams compete in pairs, and the winner of each match moves on to the next round until only one team remains. You need to generate the tournament brackets and determine the total number of matches that will be played by using recursion.

Tournament Structure:

Each round consists of half the number of teams from the previous round. The number of matches in each round is half the number of teams participating in that round. The tournament continues until there is only one team left (the champion).

Recursion Insight:

If you have n teams, you need to calculate the number of matches required to determine a winner recursively. For n teams, the number of matches required is n/2 plus the number of matches required for n/2 teams.

Q8 Tower of Hanoi is a mathematical puzzle where we have three rods (A, B, and C) and N disks. Initially, all the disks are stacked in decreasing value of diameter i.e., the smallest disk is placed on the top and they are on rod A. The objective of the puzzle is to move the entire stack to another rod (here considered C), obeying the following simple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.

Implement the solution for the tower Hanoi problem using recursion.

****