***Session 3***

**Course:** Data Structures (CS2001) **Semester:** Fall 2021

**Instructor:**  **T.A:** N/A

**Note:**

* + - * Lab manual cover following below recursion topics

**{Base Condition, Direct and Indirect Recursion, Tailed Recursion, Nested Recursion, Backtracking}**

* Maintain discipline during the lab.
* Just raise hand if you have any problem.
* Completing all tasks of each lab is compulsory.
* Get your lab checked at the end of the session.

**Base Condition in Recursion**

**Sample Code**

int Funct(int n)

{ if (n < = 1) // base case return 1;

else

return Funct (n-1);

}

**Key Points**: In the above example, base case for n < = 1 is defined and larger value of number can be solved by converting to smaller one till base case is reached.

**Task-1:**

1. Generate the following sequence with recursive approach

1 , 3 , 6 , 10 , 15 , 21 , 28 . . . .

1. Generate the following sequence with recursive approach

1 , 1 , 2 , 4 , 7 , 11 , 16 , 22 . . . .

**Direct and Indirect Recursion**

**Sample Code (Direct Recursion)**

void X()

{ // Some code....

X();

// Some code...

}

**Sample Code (In-Direct Recursion)**

void indirectRecFun1()

{ // Some code...

indirectRecFun2();

// Some code...

}

void indirectRecFun2()

{ // Some code...

indirectRecFun1();

// Some code...

}

**Task-2:**

1. **Write a indirect recursive code for the above task-1 (a,b) part with same approach as defined in the above sample code of In-Direct Recursion**

**Tailed and Non Tailed Recursion**

**Sample Code (Non tailed Recursion)**

void Funct (int a)

{

if (a < 1) return;

cout << "The current Output is " << n;

// recursive call

Funct (n-1);

}

**Sample Code (Tailed Recursion)**

unsigned Funct 1(int n, int a)

{

if (n == 1) return a;

return Funct 1(n-1, n\*a);

}

int Funct 2(unsigned int n)

{

return Funct 1(n, 1);

}

int main()

{

cout << Funct 2(5);

return 0;

}

**Task 3:**

Sort The Unsorted Numbers with both tail recursive and Normal recursive approach

**Sample Input and Output**

Given array is

12 11 13 5 6 7

Sorted array is

5 6 7 11 12 13

**Nested Recursion**

**Sample Code**

#include <iostream>

using namespace std;

int fun(int n)

{

    if (n > 100)

        return n - 10;

    // A recursive function passing parameter

    // as a recursive call or recursion inside

    // the recursion

    return fun(fun(n + 11));

}

int main()

{

    int r;

    r = fun(95);

    cout << " " << r;

    return 0;

}

**Task 4:**

**Dry run the** outputs of the upper code in order to find out how the recursive calls are made

**Backtracking**

**Sample**