

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
NITK – Surathkal**

**CS204: Data Structures and Algorithms Lab  
Assignment - 5**

**Instructions:**

1. Implement the following exercise using C.
2. You are required to complete this exercise on or before 30/12/2022.
3. Submit all the programs in a single `.zip` file.

**Exercise:**

1. Write a program to implement a complete binary tree using Linked lists. The programs should include functions to perform the following operations:
  - a. `Insert()`: inserts a new item to the complete binary tree. The items are of integer type.
  - b. `Height()`: returns the height of a node recursively.
  - c. `Preorder()`: returns the preorder traversal sequence of the binary tree. Use recursive implementation.
  - d. `Postorder()`: returns the postorder traversal sequence of the binary tree. Use recursive implementation.
  - e. `Inorder()`: returns the inorder traversal sequence of the binary tree. Use recursive implementation.
  - f. `Levelorder()`: returns the level order traversal sequence of the binary tree.
2. Write a program to implement a binary search tree (BST) having the following functionalities.
  - a. `BSTInsert()`: This function adds a given item to the BST. If the item already exists in the BST then it will not insert the ITEM anymore.
  - b. `BSTSearch()`: This function finds the location of a given item in the BST. The function also returns the parent location of the node containing the item.
  - c. `BSTInorderStack()`: This function finds the Inorder traversal sequence of a BST using stack. You are not supposed to use a recursive implementation of Inorder traversal.
  - d. `BSTDelete()`: Deletes the nodes. (Consider all the cases of nodes with degrees 0, 1, and 2).
  - e. `BST_Successor()`: finds the inorder successor of a node.
  - f. `BST_Predecessor()`: finds the inorder predecessor of the node.

- g. BST\_Findmin(): find the minimum element in a tree.
  - h. BST\_findmax(): find the maximum element in a tree.
3. Write a program to implement the Heapsort algorithm.