भारतीय सूचना प्रौद्योगिकी संस्थान गुवाहाटी

Indian Institute of Information Technology Guwahati

DATA STRUCTURES LAB (CS111)

ASSIGNMENTS–06

Assignments to be completed during lab sessions

1. Implement a stack with an array. The maximum size of the stack is user input. Your implementation should include:
    i. A structure to define the stack data type.
    ii. A pseudo-constructor function to create a new stack.
    iii. A function to check if the stack is empty.
    iv. A function to check if the stack is full.
    v. The push operation.
    vi. The pop operation.
    vii. A function to peek the item from the top.
2. Implement a circular queue with an array. The maximum size of the queue is user input. Your implementation should include:
    i. A structure to define the queue data type.
    ii. A pseudo-constructor function to create a new queue.
    iii. A function to check if the queue is empty.
    iv. A function to check if the queue is full.
    v. The enqueue operation.
    vi. The dequeue operation.
    vii. A function to get the item from the front.
    viii. A function to get the item from the rear.

3. Write functions to perform the following operations on doubly-linked lists (not circular). *You must store the information about the head and the tail nodes using two different pointers.*

   i. Write a function to add an element at the beginning of the list.

   ii. Write a function to print the elements in the list both with forward and backward traversals.

   iii. Write a function to count the number of elements in the list.

   iv. Write a function to remove the first element of the list.

   v. Write a function to add an element at the end of the list.

   vi. Write a function to remove the last element of the list.

   vii. Write a function to add an element at a given list position.

   viii. Write a function to remove the element at a given list position.

   ix. Write a function to add data after the first occurrence of a given key value in the linked list.

   x. Write a function to remove the first occurrence of a given data present in the list.

   xi. Write a function to reverse the elements in the list.

   xii. Write a function to reverse the elements in the list without creating a new list.

   xiii. Write a function to insert an element in a sorted list so the final list remains sorted.

   xiv. Write a function to sort the elements in a list.

   xv. Write a function to merge two lists.

   xvi. Write a function to get/access the data at the ith node of the list.

   xvii. Write a function to merge two sorted linked lists so the resultant list remains sorted.

   xviii. Use recursion to print the list.

   xix. Use recursion to print the list in the reverse order.

   xx. Use recursion to reverse the list.